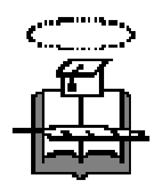


UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Físicas

Carrera de Ingeniería en Sistemas Computacionales

“Sistema de Facturación para ISP vía Web”

PROYECTO DE GRADO

CURSO DE GRADUACIÓN

Previo a la Obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTORES:

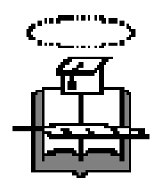
Sánchez Caicedo Bertha Verónica

Cruz Rosales Diana Elizabeth

Terán Ronquillo Pedro Pablo

GUAYAQUIL-ECUADOR

Año: 2010



UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Físicas

Carrera de Ingeniería en Sistemas Computacionales

"Sistema de Facturación para ISP vía Web"

PROYECTO DE GRADO

CURSO DE GRADUACIÓN

Previo a la Obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTORES:

Sánchez Caicedo Bertha Verónica

Cruz Rosales Diana Elizabeth

Terán Ronquillo Pedro Pablo

GUAYAQUIL-ECUADOR

Año: 2010

INDICE GENERAL

CAPITULO 1

1. Manual Técnico

1.1 Introducción

El presente documento integra las normativas de carácter interno para que el personal adscrito en el Desarrollo lleve a cabo la elaboración de documentos administrativos que faciliten la capacitación y comprensión en el uso y operación del sistema Bemusoft.

El desarrollo de Bemusoft ha permitido brindar una mejor distribución, no solo con ofrecer un buen servicio de internet sino a la hora manejar un estricto y confiado control de la facturación, como lo es la asignación exacta o correspondiente del ancho de banda de acuerdo al tipo de plan adquirido por el cliente así como también mejoras de tiempo de respuesta en el proceso de facturación.

1.2 Objetivos Generales del Sistema

Brindamos información confiable, los datos son almacenados en repositorios íntegros capaces de manejar y procesar la

información de manera óptima, manejando accesos inmediatos a la información teniendo disponibilidad de la misma.

Brindamos un sistema medible, capaz de procesar indicadores o variables para recolectar informes que permitan medir sus índices de ventas o procesos internos.

1.3 Objetivos específicos

Disponemos de un servicio personalizado pensado en las necesidades del cliente, capaz de que el sistema pueda acoplarse a las múltiples posiciones sociales o empresariales, parametrizando los planes.

Agilitamos los flujos de algunos sistemas eliminado los tiempos de Improductividad que conlleven a un retardo en el procesamiento de la información, generando concurrencia que no pueda ser atendida a tiempo.

Nuestro sistema y personal está en la capacidad de brindar o crear reglas de negocio para lograr o crear estándares aplicables en los múltiplos usuarios y tener un mejor control sobre los procesos.

Brindamos escalabilidad, capaz de brindar servicios aun en mercados en crecimiento.

CAPITULO 2

2. Contenido técnico

2.1 Reglas del Negocio

- El cliente (ISP) podrá parametrizar:
- Activación/inactivación del usuario, además de poder Resetear la claves.
- Optar por generar o ingresar manualmente el código de los productos (Planes de internet), y podrá decidir que prefijo acompaña el código automático.
- Ingresar el ciclo de facturación que mejor se adapte a sus necesidades
- Ingresar el rango de facturas físicas para generar facturas informativas para sus clientes, cabe recalcar que dichas facturas no tienen validez jurídica.
- El sistema realiza una facturación basado en impuestos vigentes, si en el futuro se incluyera otro impuesto por cuestiones gubernamentales, deberán contactarse con Bemusoft S.A. para su respectiva modificación.
- Como regla se ofrecerá un plan por contrato, y el servicio a un ordenador por plan.

- El cliente final podrá renunciar al servicio de internet si y solo si no mantiene deuda alguna con su ISP, en caso de que renuncie antes de terminado el contrato tendría que pagar el costo prorrateado por los días que gozo de el servicio.
- El en caso de que algún usuario dejase de laborar para el ISP mediante nuestra aplicación solo podrá inactivarlo mas no borrarlo.
- Un usuario solo podrá tener un rol.
- El administrador podrá agregar más opciones a los roles si así lo deseara.
- El día de corte de la factura se enviara un mail al cliente final informándole el valor que debe cancelar, solo dentro de su dominio.
- El corte se realizara cinco días después de la fecha de corte, procurando darle días de gracia usándolo como una estrategia para fidelizar clientes.

CAPITULO 3

3. Herramientas y configuraciones

A continuación mencionaremos las herramientas de software necesarias para el desarrollo de la aplicación:

- IDE MyEclipse 5
- Java 5
- Tomcat 5
- MySQL 5
- Linux
- Centos 5
- DHCP
- CBQ
- SendMail
- Iptables

3.1 MyEclipse.-

MyEclipse es un IDE creado y mantenido por la compañía Genuitec, un miembro fundador del eclipse.

MyEclipse se construye sobre plataforma Eclipse, e integra el propietario y soluciones abiertas de la fuente en el ambiente del desarrollo.

3.2 Tomcat.-

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

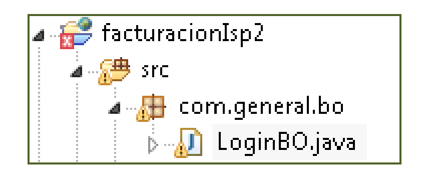
3.3 MySQL.-

El software MySQL® proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multi-threaded, multi usuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido.

El software tiene una doble licencia. Los usuarios pueden elegir entre usar el Software MySQL como un producto Open Source bajo los términos de la licencia GNU General Public License (http://www.fsf.org/licenses/) o pueden adquirir una licencia comercial estándar de MySQL AB.

CAPITULO 4

4.1 Código Fuente




```
Package com.general.bo;
/* CLASE QUE  CONTIENE TODAS LAS VALIDACIONES
 * PARA LA VALIDACION DEL USUARIO
 * VERIFICA USUARIO Y PASSWORD
 * */
import java.io.UnsupportedEncodingException;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.spec.InvalidKeySpecException;

import javax.crypto.BadPaddingException;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;

import com.general.modelo.persistentes.*;
import com.general.util.*;

public class LoginBO implements Configuracion{

        public   Usuarios   validarLoginReintentos(String
ps_user, String ps_password) throws Exception
        {
            try
            {
                    Usuarios
l_usuario=validarLogin(ps_user.toUpperCase(),
ps_password);
                  return l_usuario;
            }
            catch (Exception e)
            {
                    throw e;
            }
        }
```

```
        public   Usuarios   cambiarClave(String   ps_user,
String  ps_claveActual,  String  ps_claveNueva)  throws
Exception
        {
            ps_user=ps_user.toUpperCase();
            ps_claveActual=ps_claveActual;
            ps_claveNueva=ps_claveNueva;
            try
            {
                  DAOUtil.beginTransaction();
                  Usuarios l_usuario=null;
                  l_usuario=validarLogin(ps_user,
ps_claveActual);
                  if(ps_claveActual.equals(ps_claveNueva))
                  {
                        ExcepcionLogicaNegocio      l_e=new
ExcepcionLogicaNegocio();
                        l_e.setMensajeUsuario("La     Clave
Nueva debe ser diferente a la Clave Actual");

        l_e.setMensajeTecnico("ps_claveActual.equals(ps_c
laveNueva)");
                        throw l_e;
                  }
                  l_usuario.setCambioClave("S");

        l_usuario.setClave(UtilCryptography.encriptar(ps_
claveNueva));
                  //l_usuario.setActualizarClave("N");
                  UsuariosDAO l_uHome=new UsuariosDAO();
                  l_uHome.attachDirty(l_usuario);
                  DAOUtil.commit();
                  return l_usuario;
            }
            catch (Exception e)
            {
                  DAOUtil.rollback();
                  throw e;
            }
        }

        private   Usuarios   validarLogin(String   ps_user,
String       ps_password)       throws       Exception,
InvalidKeyException,            InvalidKeySpecException,
NoSuchAlgorithmException,        NoSuchPaddingException,
InvalidAlgorithmParameterException,
IllegalBlockSizeException,          BadPaddingException,
UnsupportedEncodingException, ExcepcionLogicaNegocio
        {
```

```
            /*
             *  CONTIENE  LA  VALIDACION  DEL  USER  Y  DEL
PASSWORD
             * */
            UsuariosDAO l_home=new UsuariosDAO();
            Usuarios
l_usuario=(Usuarios)l_home.findByUser(ps_user);

//ps_password=ps_password.toUpperCase().trim();      --No
debo hacerle upper case a la clave

ps_password=UtilCryptography.encriptar(ps_password);
             if(l_usuario==null)
             {
                  throw  new  Exception("No  se  encontro  el
usuario " + ps_user.toUpperCase() + " registrado en el
sistema");
             }
             if(l_usuario.getEstado().equals("I"))
             {
                  throw   new   Exception("El   usuario   "   +
ps_user.toUpperCase()  +  "  se  encuentra  en  un  estado
invalido registrado en el sistema");
             }
             if(!l_usuario.getClave().equals(ps_password))
             {
                  throw   new   Exception("El   usuario   "   +
ps_user.toUpperCase()  +  "  tiene  una  clave  diferente
registrada en el sistema");
             }
             //l_usuario.setReintentosClave((short) 0);
             return l_usuario;
        }
}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
InicializadorContexto.java
ManejadorHibernateRequest.java

```
package com.general.escuchadores;
/*
 * Inicializa los log de la aplicacion de auditoria
 * registra en la consola de la aplicacion cuando inicia
 * y cuando termina la aplicacion
 * */
```

```
import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;
import org.apache.log4j.Level;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;

public       class       InicializadorContexto       implements
ServletContextListener
{
       static            Logger            logger             =
Logger.getLogger(InicializadorContexto.class);
       public void contextDestroyed(ServletContextEvent p_sc)
       {
          System.out.println("Destrucion   de   la   aplicacion
web");
          if(logger.isEnabledFor(Level.INFO))
               logger.info("Destruyendo      la      aplicacion
"+p_sc.getServletContext().getServletContextName());
       }
       public void contextInitialized(ServletContextEvent p_sc)
       {

       PropertyConfigurator.configure(p_sc.getServletContext().
getRealPath("/WEB-INF/log4j.properties"));
          try
          {
               System.out.println("Inicio   de   la   aplicacion
web");
               if(logger.isEnabledFor(Level.INFO))
               {
                    logger.info("Inicializado   la   aplicacion
"+p_sc.getServletContext().getServletContextName());
               }
          }
          catch (Exception e)
          {
               e.printStackTrace();
          }
       }
}
```

```
package com.general.escuchadores;
/*
 * Incialmente comprueba si hay comunicacion con la base de
datos
 * por medio de la clase DAOUTIl usando hibernate
 *
 * considerando que si no esta subida la base de datos no
```

```
sube la aplicacion
 * */
import javax.servlet.ServletRequestEvent;
import javax.servlet.ServletRequestListener;
import org.apache.log4j.Logger;
import com.general.util.DAOUtil;

public class ManejadorHibernateRequest implements
ServletRequestListener
{
       static Logger logger =
Logger.getLogger(InicializadorContexto.class);
       public void requestDestroyed(ServletRequestEvent
p_sre)
       {
          try
          {
               DAOUtil.rollback();
          }
          catch (Exception e)
          {
          }
       }

       public void requestInitialized(ServletRequestEvent
arg0)
       {
          DAOUtil.beginTransaction();
       }
}
```

com.general.filtros.generales
FilSensarUser.java

```
package com.general.filtros.generales;

import java.io.IOException;
import java.util.Date;

import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import com.general.modelo.persistentes.Usuarios;
import com.general.util.Sesion;

public class FilSensarUser implements Filter{
          private FilterConfig filterConfig = null;

          public void init(FilterConfig filterConfig)throws
ServletException {
               this.filterConfig = filterConfig;
          }

          public        void        doFilter(ServletRequest
request,ServletResponse  response,FilterChain  chain)  throws
ServletException {
               try {
                    Log                log                =
LogFactory.getLog(FilSensarUser.class);
                    log.debug("EJECUCION     DE     FILTRO
GENERAL");
                    System.out.println("Inicio    Ejecucion
De Filtro General " + this.getClass() + " - "+ new Date());
                    HttpServletRequest        re         =
(HttpServletRequest) request;
                    HttpServletResponse       rere       =
(HttpServletResponse) response;
                    HttpSession        sesionWeb         =
re.getSession();
                    String nombreUri = re.getRequestURI();
               String nombreContext = re.getContextPath();
               String nombrePaginaActual = "";
               String             rutaInicial             =
"http://"+re.getRemoteAddr()+":"+re.getServerPort()+re.getCo
ntextPath();
               nombreUri = nombreUri.replaceAll("/","");
               nombreContext                              =
nombreContext.replaceAll("/", "");
                    System.out.println("URI:      "      +
nombreUri);
                    System.out.println("CONTEXT:    "    +
nombreContext);



                    if (sesionWeb == null)
               {
               log.debug("Session  caducada  no  se  puede
```

```
ingresar a la opcion" );
                rere.getWriter().println("          <script>
top.close(); window.open('"+ rutaInicial +"'); </script>");
                }
if(nombreUri.toUpperCase().equals(nombreContext.toUpperCase(
)))
                     {
                          System.out.println("La  sesion  es
nueva continuar");
                          log.debug("La   sesion   es   nueva
continuar" );
                     }
                     else
                     {
                          nombrePaginaActual                =
nombreUri.substring((nombreContext.length()));
                          System.out.println("La  sesion  es
antigua");
                          log.debug("Validacion de usuario"
);
                          if
(nombrePaginaActual.toUpperCase().indexOf("LOGIN") != -1 ||

       nombrePaginaActual.toUpperCase().indexOf("ERROR")  !=
-1
                              )
                          {
                               System.out.println("Es    el
login continuar");
                               log.debug("Es    el    login
continuar" );
                               rere.getWriter().println("
<script>  top.close();   window.open('"+  rutaInicial  +"');
</script>");
                          }
                          else
                          {
                               Usuarios     l_usuario     =
(Usuarios)sesionWeb.getAttribute(Sesion.USUARIO);
                               if (l_usuario == null )
                               {
                                    log.debug("El   usuario
no esta logoneado, por favor reingrese" );

       rere.getWriter().println("   <script>   top.close();
window.open('"+ rutaInicial +"'); </script>");
                               }
                          }
                     }
```

```
                System.out.println("Fin Ejecucion De Filtro
General " + this.getClass() + " - "+ new Date());
                     chain.doFilter(re,rere);


                } catch (IOException e) {
                     // TODO Auto-generated catch block
                     e.printStackTrace();
                }
           }
           public void destroy(){
                this.filterConfig =null;   }
```

com.general.struts.action
- ACambiarClave.java
- ActivarImpuestoAction.java
- ActivarRolAction.java
- ActivarUsuarioAction.java
- ActualizarParametrosAction.java

```
/*
 * Generated by MyEclipse Struts
 * Template path: templates/java/JavaClass.vtl
 */
package com.general.struts.action;

import java.util.Date;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.general.modelo.persistentes.Parametros;
import com.general.modelo.persistentes.ParametrosDAO;
import com.general.modelo.persistentes.Usuarios;
import com.general.struts.form.ActualizarParametrosForm;
import com.general.util.DAOUtil;
import com.general.util.Sesion;

/**
 * MyEclipse Struts
 * Creation date: 08-02-2007
 *
```

```
 * XDoclet definition:
 *        @struts.action        path="/actualizarParametros"
name="actualizarParametrosForm"
input="/paginas/transacciones/seguridad/actualizarParametro.
jsp" scope="request" validate="true"
 *     @struts.action-forward     name="ok"     path="sdfdfdf"
redirect="true"
 *     @struts.action-forward    name="fallo"    path="dsfsd"
redirect="true"
 */
public class ActualizarParametrosAction extends Action {
	/*
	 * Generated Methods
	 */

	/**
	 * Method execute
	 * @param mapping
	 * @param form
	 * @param request
	 * @param response
	 * @return ActionForward
	 */
	public  ActionForward  execute(ActionMapping  mapping,
ActionForm form,
			HttpServletRequest                     request,
HttpServletResponse response) {
		ActualizarParametrosForm  actualizarParametrosForm
=  (ActualizarParametrosForm)  form;//  TODO  Auto-generated
method stub
		try
		{
			if
(actualizarParametrosForm.getValorParametro().length() < 1)
			{
				throw  new  Exception("Debe  ingresar  un
valor para el parametro");
			}
			DAOUtil.beginTransaction();
			ParametrosDAO      parameDao      =      new
ParametrosDAO();
			Parametros   param   =   parameDao.findById(new
Integer(actualizarParametrosForm.getIdParametro()));

	param.setValorParametro(actualizarParametrosForm.getV
alorParametro().toUpperCase());

			if  (param.getDescripcion().equals("IVA")  ||
param.getDescripcion().equals("ICE") )
```

```
                {
                     try
                     {

        Float.parseFloat(actualizarParametrosForm.getValorPar
ametro());

                     }catch (Exception e)
                     {
                          throw new Exception("El valor del
parametro debe ser numerico, para el impuesto de facturas");
                     }
                }

                //param.setObservacion("Actualizacion      de
parametro");

        param.setUsuarioIngreso(((Usuarios)request.getSession
().getAttribute(Sesion.USUARIO)).getDescUsuario());
                param.setFechaActualizacion(new Date());
                parameDao.attachDirty(param);
                DAOUtil.commit();

        request.getSession().setAttribute("MENSAJEOK",       "
Actualizacion De Parametro Exitoso");
                request.getSession().setAttribute("PAGINA",
"/filConfigtacion.do");
                return mapping.findForward("ok");
           }catch(Exception e)
           {
                DAOUtil.rollback();
                e.printStackTrace();

        request.getSession().setAttribute("ERRORGENERAL", e);

        request.getSession().setAttribute("TITULOERROR",
"Error Actualizacion Parametro");
                return mapping.findForward("fallo");
           }
        }
}
```

```
/*
 * Generated by MyEclipse Struts
 * Template path: templates/java/JavaClass.vtl
 */
package com.general.struts.action;
```

```
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Set;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.general.modelo.persistentes.Clientes;
import com.general.modelo.persistentes.Personas;
import com.general.modelo.persistentes.PersonasDAO;
import com.general.modelo.persistentes.Usuarios;
import com.general.modelo.persistentes.UsuariosDAO;
import com.general.util.DAOUtil;
import com.general.util.Operaciones;
import com.general.util.Sesion;
import com.general.util.ValidacionesVarias;

/**
 * MyEclipse Struts
 * Creation date: 08-29-2009
 *
 * XDoclet definition:
 * @struts.action validate="true"
 * @struts.action-forward name="procesoFallido" path="sdsd"
redirect="true"
 * @struts.action-forward name="procesoExitoso" path="sdsd"
redirect="true"
 */
public class BuscarIdentificacionClienteAction extends
Action {
        /*
         * Generated Methods
         */

        /**
         * Method execute
         * @param mapping
         * @param form
         * @param request
         * @param response
         * @return ActionForward
         */
        public ActionForward execute(ActionMapping mapping,
ActionForm form,
```

```
                HttpServletRequest                  request,
HttpServletResponse response) {
           // TODO Auto-generated method stub
           String principalError = "Error Crear Rol: ";
           String           datoIdentificacion          =
request.getParameter("datoIdentificacion");
           String accion = request.getParameter("accion");
           String lsGeneraUsuario = "";
           boolean bandera = false;
           try
           {
                String aux = request.getParameter("aux");
                accion = accion==null?"0":accion;
                request.getSession().setAttribute("ACCION",
accion);

        request.getSession().setAttribute("IDENTIFICACION",
datoIdentificacion);
                request.getSession().setAttribute("TIPO",
aux);

        request.getSession().setAttribute("TITULOERROR",
null);

        request.getSession().setAttribute("ERRORGENERAL",
null);
                if (datoIdentificacion.trim().equals(""))
                {
                     throw  new  Exception  ("Ingrese  una
identificación para realizar la busqueda");
                }

                if (datoIdentificacion.trim().length() < 10)
                {
                     throw     new     Exception     ("La
identificacion debe tener mas de 10 caracteres");
                }

                HashMap                mapa                =
Operaciones.validaDeudaCliente(datoIdentificacion);
                Double deuda = (Double) mapa.get("DEUDA");
                double deudafinal  = deuda.doubleValue();
                String      observacion      =      (String)
mapa.get("OBSERVACION");
                if (deudafinal > 0)
                {
                     throw new Exception("No se puede crear
el contrato, motivo: " + observacion.toUpperCase());
                }
```

```
                DAOUtil.beginTransaction();
                PersonasDAO personaDao = new PersonasDAO();
                Personas              persona              =
personaDao.findByPropertyUnique(personaDao.CEDULA,
datoIdentificacion);

                if (!datoIdentificacion.trim().equals(""))
                {

        ValidacionesVarias.validarDocumento(datoIdentificaci
on);
                }

        request.getSession().setAttribute("IDENTPER",datoIde
ntificacion);

                if (persona == null)
                {
                     //Buscar por ruc
                     persona                                =
personaDao.findByPropertyUnique(personaDao.RUC,
datoIdentificacion);
                     if (persona == null)
                     {
                          persona                           =
personaDao.findByPropertyUnique(personaDao.PASAPORTE,
datoIdentificacion);
                          if (persona==null)
                          {
                               if (accion.equals("1"))
                               {

        request.getSession().setAttribute("IDENTIFICACIONINI
CIO", datoIdentificacion);

        request.getSession().setAttribute("ASIG", "1");
                                    return
mapping.findForward("busquedaFallida");
                               }
                               throw  new  Exception("No  se
encontro persona registrada con el numero de identificación
" + datoIdentificacion);
                          }
                     }
                }
                //Si encontro persona ahora buscar cliente
                Set clientes = persona.getClienteses();
              Clientes cli = null;
```

```
               Iterator ite = clientes.iterator();

               while(ite.hasNext())
               {
                cli = (Clientes)ite.next();
               }

               //Si el cliente no existe debe ir a crearlo
siempre si importar la opcion
               if (cli == null)
               {
                //Crear el cliente sin importar nada

       request.getSession().setAttribute("PERSONAENCONTRADA
",persona);

       request.getSession().setAttribute("CLIENTEENCONTRADO
",cli);

                return
mapping.findForward("procesoExitoso");
               }

                DAOUtil.commit();


       request.getSession().setAttribute("PERSONAENCONTRADA
",persona);

       request.getSession().setAttribute("CLIENTEENCONTRADO
",cli);
                if (accion.equals("1"))
                {
                     return
mapping.findForward("busquedaExitosa");
                }
                else
                {
                     return
mapping.findForward("procesoExitoso");
                }

           }catch (Exception e)
           {
                e.printStackTrace();

       request.getSession().setAttribute("TITULOERROR",
"Busqueda Persona");
```

```
        request.getSession().setAttribute("ERRORGENERAL",
e);//new Exception(principalError+e.getMessage()));
                if (accion.equals("1"))
                {
                     return
mapping.findForward("busquedaError");
                }
                else
                {
                     return
mapping.findForward("procesoFallido");
                }
           }
        }
}
```

```
/*
 * Generated by MyEclipse Struts
 * Template path: templates/java/JavaClass.vtl
 */
package com.general.struts.action;

import java.util.Iterator;
import java.util.List;
import java.util.Set;
import java.util.Vector;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.general.modelo.persistentes.Opciones;
import com.general.modelo.persistentes.OpcionesDAO;
import com.general.modelo.persistentes.OpcionesRoles;
import com.general.modelo.persistentes.Roles;
import com.general.modelo.persistentes.RolesDAO;
import com.general.util.DAOUtil;

/**
 * MyEclipse Struts
 * Creation date: 08-02-2007
 *
 * XDoclet definition:
 * @struts.action validate="true"
 *              @struts.action-forward             name="ok"
```

```
path="/paginas/transacciones/seguridad/gestionOpcionesRoles.
jsp" redirect="true"
 *            @struts.action-forward            name="fallo"
path="/paginas/transacciones/seguridad/gestionOpcionesRoles.
jsp" redirect="true"
 */
public class BuscarOpcionesModAction extends Action {
	/*
	 * Generated Methods
	 */

	/**
	 * Method execute
	 * @param mapping
	 * @param form
	 * @param request
	 * @param response
	 * @return ActionForward
	 */
	public  ActionForward  execute(ActionMapping  mapping,
ActionForm form,
			HttpServletRequest                  request,
HttpServletResponse response) {
		// TODO Auto-generated method stub
		try
		{
			Roles                  rol                   =
(Roles)request.getSession().getAttribute("ROL_ENC");
			String               idOpcion                =
request.getParameter("idOpcion");
			DAOUtil.beginTransaction();
			RolesDAO rolDao = new RolesDAO();
			Roles                  rol1                  =
rolDao.findById(rol.getIdRol());
			OpcionesDAO opcDao = new OpcionesDAO();
			Opciones    opc    =    opcDao.findById(new
Integer(idOpcion));

			Set opcionesRol = rol1.getOpcionesRoleses();
			Iterator totalOpc = opcionesRol.iterator();

			//obtener todas las opciones activas
			List           opcionesActivas               =
opcDao.findByEstado("A");
			Iterator            opcActivas               =
opcionesActivas.iterator();
			Vector opcionesAsignadas = new Vector();
			Vector opcionesNoAsignadas = new Vector();
```

```
                Set opcionesDelModulo = opc.getOpcioneses();

                //Barrerme los registros del modulo
                Iterator               iter               =
opcionesDelModulo.iterator();
                boolean guardaAsigando = false;
                while (iter.hasNext())
                {
                     Opciones opce = (Opciones)iter.next();
                     Set             opcRoles             =
opce.getOpcionesRoleses();
                     Iterator ter2 = opcRoles.iterator();
                     guardaAsigando = false;
                     while(ter2.hasNext())
                     {
                          OpcionesRoles       oprR       =
(OpcionesRoles)ter2.next();
                          if
(oprR.getRoles().getIdRol().toString().equals(rol1.getIdRol(
).toString()))
                          {
                               //Debo   asignarlo   a   los
asignados
                               guardaAsigando = true;
                          }
                     }
                     if (guardaAsigando)
                     {
                          opcionesAsignadas.add(opce);
                     }else
                     {
                          opcionesNoAsignadas.add(opce);
                     }

                }

                /*
                while (totalOpc.hasNext())
                {
                     OpcionesRoles        opcRoles        =
(OpcionesRoles) totalOpc.next();
                     Opciones           opcion            =
opcRoles.getOpciones();
                     if (opcion.getOpciones()!=null)
                     {
                          if
(opcion.getOpciones().getIdOpcion().toString().equals(opc.ge
tIdOpcion().toString()))
                          {
```

```
                                //Aqui   estoy   sacando   las
opciones que estan asigandas al rol
                                //y  que  tienen  como  modulo
padre a la opcion que seleccione
                                if
(opcRoles.getRoles().getIdRol().toString().equals(rol1.getId
Rol().toString()) && opcRoles.getEstado().equals("A"))
                                {
                                     //asignadas

     opcionesAsignadas.add(opcion);

                                }
                                else
                                {
                                     //no asignadas

     opcionesNoAsignadas.add(opcion);
                                }
                           }
                      }
                 }
                 //Debo sumar las opciones que no estan en la
de opciones roles
                 List             totalOpciones             =
opcDao.findByEstado("A");

                 Iterator         totalOpcionesSin          =
totalOpciones.iterator();
                 Set              opcionesRol1              =
rol1.getOpcionesRoleses();
                 Iterator totalOpc1 = opcionesRol.iterator();
                 while (totalOpcionesSin.hasNext())
                 {
                      Opciones    opcionSin    =    (Opciones)
totalOpcionesSin.next();
                      try
                      {
                      if
(opcionSin.getOpciones().getIdOpcion().toString().equals(opc
.getIdOpcion().toString()))
                      {
                           if
(opcionSin.getOpcionesRoleses().size() == 0)
                           {

     opcionesNoAsignadas.add(opcionSin);
                           }
                      }
```

```
                    }catch (Exception eFin)
                    {
                         ;
                    }
               }
               */
               request.getSession().setAttribute("OPC_ENC",
opc);

     request.getSession().setAttribute("OPC_MOD_ASIG",
opcionesAsignadas);

     request.getSession().setAttribute("OPC_MOD_NO_ASIG",
opcionesNoAsignadas);
               DAOUtil.commit();
               return mapping.findForward("ok");
          }catch(Exception e)
          {
               e.printStackTrace();
               DAOUtil.rollback();

     request.getSession().setAttribute("TITULOERROR",
"Buscar Opciones");

     request.getSession().setAttribute("ERRORGENERAL",
e);//new Exception(principalError+e.getMessage()));
               return mapping.findForward("fallo");
          }
     }
}
```

```
/*
 * Generated by MyEclipse Struts
 * Template path: templates/java/JavaClass.vtl
 */
package com.general.struts.action;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.Vector;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
```

```
import com.general.modelo.persistentes.CiclosFacturaciones;
import
com.general.modelo.persistentes.CiclosFacturacionesDAO;
import com.general.modelo.persistentes.Clientes;
import com.general.modelo.persistentes.Contratos;
import com.general.modelo.persistentes.ContratosDAO;
import com.general.modelo.persistentes.EmpleadosDAO;
import com.general.modelo.persistentes.Personas;
import com.general.modelo.persistentes.PlazosPlznesDAO;
import com.general.modelo.persistentes.Productos;
import com.general.modelo.persistentes.ServiciosContratados;
import
com.general.modelo.persistentes.ServiciosContratadosDAO;
import com.general.modelo.persistentes.Usuarios;
import com.general.util.DAOUtil;
import com.general.util.Sesion;

/**
 * MyEclipse Struts
 * Creation date: 08-02-2007
 *
 * XDoclet definition:
 * @struts.action validate="true"
 */
public class CreacionContrato extends Action {
	/*
	 * Generated Methods
	 */

	/**
	 * Method execute
	 * @param mapping
	 * @param form
	 * @param request
	 * @param response
	 * @return ActionForward
	 */
	public ActionForward execute(ActionMapping mapping,
ActionForm form,
		HttpServletRequest request,
HttpServletResponse response) {
		// TODO Auto-generated method stub
		try
		{
			Personas persona =
(Personas)request.getSession().getAttribute("PERSONAENCONTRA
DA");
			Clientes cliente =
```

```
(Clientes)request.getSession().getAttribute("CLIENTEENCONTRA
DO");
                Vector                planes                =
(Vector)request.getSession().getAttribute("CARRITOCOMPRAS");
                //CiclosFacturacionesDAO ciclo =
                String               idCiclo                =
request.getParameter("ciclo");
                String             observacion              =
request.getParameter("observacion");
                //Eso necesito

                if (persona==null)
                {
                     throw  new  Exception("No  se  encuentra
la persona en la session, por favor revisar");
                }

                if (cliente==null)
                {
                     throw  new  Exception("No  se  encuentra
el cliente en la session, por favor revisar");
                }

                if (planes==null)
                {
                     throw  new  Exception("No  se  encuentra
los planes seleccionados, por favor revisar");
                }

                if (planes.size()==0)
                {
                     throw     new     Exception("No     ha
seleccionado planes, por favor revisar");
                }

                if (idCiclo==null)
                {
                     throw   new   Exception("Seleccione   el
ciclo por favor");
                }

                if (idCiclo.equals("-1"))
                {
                     throw   new   Exception("Seleccione   el
ciclo por favor");
                }


                if (observacion==null)
```

```
                {
                     throw    new    Exception("Ingrese    la
observacion, por favor");
                }

                if (observacion.trim().equals(""))
                {
                     throw    new    Exception("Ingrese    la
observacion, por favor");
                }
            ///Realizar   la   transaccion   de   creacion   de
contrato y servicio
                DAOUtil.beginTransaction();

                Contratos             contrato             =
creacionContrato(persona,cliente,      planes,      idCiclo,
observacion,
((Usuarios)request.getSession().getAttribute(Sesion.USUARIO)
).getDescUsuario() );

                DAOUtil.commit();


        request.getSession().setAttribute("CONTRATOACTUAL",
contrato);
            ///Realizar   la   transaccion   de   creacion   de
contrato y servicio

                return
mapping.findForward("procesoExitoso");
           }catch(Exception e)
           {
                DAOUtil.rollback();
                e.printStackTrace();

        request.getSession().setAttribute("TITULOERROR",
"Creacion De Contrato");

        request.getSession().setAttribute("ERRORGENERAL",
e);//new Exception(principalError+e.getMessage()));
                return
mapping.findForward("procesoFallido");
           }
        }

        private Contratos creacionContrato(Personas persona,
Clientes  cliente,  Vector  planes,  String  idCiclo,  String
observacion, String user)
        throws Exception
```

```
        {
           // TODO Auto-generated method stub
           Productos prod = (Productos)planes.elementAt(0);
           Date fecha = new Date();
           SimpleDateFormat        lsdf        =        new
SimpleDateFormat("ddMMyyyyhhmmss");
           Contratos cont = new Contratos();
           ContratosDAO contDao = new ContratosDAO();
           CiclosFacturacionesDAO     cicloDAO     =     new
CiclosFacturacionesDAO();
           CiclosFacturaciones ciclo = cicloDAO.findById(new
Integer(idCiclo));

           if (ciclo==null)
           {
                throw new Exception("No se encontro el ciclo
" + idCiclo + ", en el sistema");
           }
           EmpleadosDAO empDao = new EmpleadosDAO();

           cont.setCiclosFacturaciones(ciclo);
           cont.setClientes(cliente);

        cont.setCuenta(prod.getDescripcion().toUpperCase());
           cont.setEmpleados(null);
           cont.setEstado("R");//contrato   que   no   tiene
servicios activos primera vez
           cont.setFechaInicio(new Date());
           cont.setFechaRegistro(new Date());
           //Calcular la fecha fin = fecha inicio + plazo
           PlazosPlznesDAO plazoDao = new PlazosPlznesDAO();
           String      plazo      =      (plazoDao.findById(
prod.getPlazo())).getDescPlazo();
           cont.setPlazo( plazo );

           Calendar calendar = Calendar.getInstance();
           calendar.add(Calendar.MONTH                      ,
Integer.parseInt(plazo));


           cont.setFechaFin(calendar.getTime());

           cont.setObservacion(observacion.toUpperCase());
           //Obtener el plazo luego el precio

           cont.setPrecioPlan(new
Double(prod.getTarifaBasica().doubleValue()));
           cont.setUsuarioIngreso(user);
           cont.setValorReferencia1("Cantidad Servicios: " +
```

```
((Productos)planes.elementAt(0)).getTiposProdcutos().getCaon
tidadMaxima());

          contDao.save(cont);
          ServiciosContratadosDAO     servDao     =     new
ServiciosContratadosDAO();
          //---------------------
          //Registrar los servicios por cada plan uno mas
para el correo
          //---------------------

          for           (int           i=0;           i<
prod.getTiposProdcutos().getCaontidadMaxima().intValue();
i++)
          {

               ServiciosContratados     serv     =     new
ServiciosContratados();
               serv.setContratos(cont);

       serv.setDescripcionServicio(prod.getDescripcion());
               serv.setEstado("R");//Reservado
               serv.setFechaRegistro(new Date());
               serv.setObservacion(observacion);
               serv.setProductos(prod);
               serv.setUsuarioIngreso(user);
               serv.setValorReferencia3("PLAN INTERNET");
               serv.setFechaInicio(new Date());
               servDao.save(serv);

               String             referencia             =
"";//persona.getPrimerNombre()+"."+persona.getSegundoNombre(
)+"."+persona.getApellidoPaterno()+"."+persona.getApellidoMa
terno();
               //Generar referencia ojo con el usuario asi
como lo haces para crear el usuario
               referencia                                =
persona.getPrimerNombre().substring(0,
1)+persona.getApellidoPaterno();

               //List             referenci              =
servDao.findByValorReferencia2(referencia);

               /*if (referenci.size()==0)
               {*/
                  if (i < 1)
                  {
                    ServiciosContratados   correo   =   new
```

```
ServiciosContratados();
                     correo.setContratos(cont);

        correo.setDescripcionServicio("CORREO");
                     correo.setEstado("R");//Reservado
                     correo.setFechaRegistro(new Date());
                     correo.setObservacion(observacion);

                     correo.setUsuarioIngreso(user);
                     correo.setProductos(prod);
                     correo.setValorReferencia3("CORREO");
                     correo.setFechaInicio(new Date());
                     servDao.save(correo);
                   }
                //}
           }



           return cont;
        }
}
```

```
/*
 * Generated by MyEclipse Struts
 * Template path: templates/java/JavaClass.vtl
 */
package com.general.struts.action;

import java.io.UnsupportedEncodingException;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.spec.InvalidKeySpecException;
import java.util.Date;
import java.util.List;
import java.util.Set;

import javax.crypto.BadPaddingException;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
```

```
import com.general.modelo.persistentes.Personas;
import com.general.modelo.persistentes.PersonasDAO;
import com.general.modelo.persistentes.RolesDAO;
import com.general.modelo.persistentes.Usuarios;
import com.general.modelo.persistentes.UsuariosDAO;
import com.general.struts.form.IngresoUsuarioForm;
import com.general.util.DAOUtil;
import com.general.util.Sesion;
import com.general.util.UtilCryptography;

/**
 * MyEclipse Struts
 * Creation date: 08-29-2009
 *
 * XDoclet definition:
 *          @struts.action          path="/ingresoUsuario"
name="ingresoUsuarioForm"
input="/paginas/transacciones/seguridad/ingresoUsuario.jsp"
scope="request" validate="true"
 *       @struts.action-forward       name="procesoFallido"
path="/paginas/transacciones/seguridad/ingresoUsuario.jsp"
redirect="true"
 *       @struts.action-forward       name="procesoExitoso"
path="sdsdsd" redirect="true"
 */
public class IngresoUsuarioAction extends Action {
       /*
        * Generated Methods
        */

       /**
        * Method execute
        * @param mapping
        * @param form
        * @param request
        * @param response
        * @return ActionForward
        */
       public  ActionForward  execute(ActionMapping  mapping,
ActionForm form,
                HttpServletRequest                  request,
HttpServletResponse response) {
          IngresoUsuarioForm      ingresoUsuarioForm      =
(IngresoUsuarioForm) form;// TODO Auto-generated method stub
          String principalError = "Error Crear Rol: ";
          try
          {

       request.getSession().setAttribute("TITULOERROR",
```

```
null);

        request.getSession().setAttribute("ERRORGENERAL",
null);
                 if
(ingresoUsuarioForm.getIdPersona().trim().equals(""))
                 {
                       throw new Exception ("Debe ingresar a
la persona que se le asignará el usuario");
                 }

                 if
(ingresoUsuarioForm.getDescripcionUsuario().trim().equals("-
1"))
                 {
                       throw   new   Exception   ("Ingrese   el
nombre del usuario");
                 }

                 if
(ingresoUsuarioForm.getRol().trim().equals("-1"))
                 {
                       throw  new  Exception  ("Seleccione  el
rol del usuario");
                 }

                 if
(ingresoUsuarioForm.getObservacion().trim().equals(""))
                 {
                       throw  new  Exception  ("Seleccione  el
rol del usuario");
                 }

                 DAOUtil.beginTransaction();
                 crearUsuario(ingresoUsuarioForm,
((Usuarios)request.getSession().getAttribute(Sesion.USUARIO)
).getDescUsuario());
                 DAOUtil.commit();
                 request.getSession().setAttribute("PAGINA",
"/paginas/transacciones/seguridad/ingresoUsuario.jsp");

        request.getSession().setAttribute("MENSAJEOK",      "
Creacion De Usuario Exitoso, la clave inicial del usuario es
del 1 al 8, no olvidarse realizar el cambio de clave");
                 return
mapping.findForward("procesoExitoso");

            }catch (Exception e)
            {
```

```
        request.getSession().setAttribute("TITULOERROR",
"Creacion Usuario");

        request.getSession().setAttribute("ERRORGENERAL",
e);//new Exception(principalError+e.getMessage()));
                return
mapping.findForward("procesoFallido");
           }

        }

        private     void     crearUsuario(IngresoUsuarioForm
ingresoUsuarioForm, String descUsuario) throws Exception{
           // TODO Auto-generated method stub
           UsuariosDAO usuarioDao = new UsuariosDAO();
           Usuarios    usuario =  new Usuarios();
           PersonasDAO personaDao = new PersonasDAO();
           RolesDAO    rolesDao = new RolesDAO();
           //Validar el usuario
           List                  usuasrios                  =
usuarioDao.findByDescUsuario(ingresoUsuarioForm.getDescripci
onUsuario().toUpperCase());

           if (usuasrios.size() > 0 )
           {
                throw  new  Exception("El  usuario  "    +
ingresoUsuarioForm.getDescripcionUsuario().toUpperCase()   +
", ya se encuentra registrado por favor asignar otro");

           }
           Personas    pers    =    personaDao.findById(new
Integer(ingresoUsuarioForm.getIdPersona()));
           Set users = pers.getUsuarioses();

           if (users.size() > 0)
           {
                throw  new  Exception("La  persona  "    +
pers.getPrimerNombre() + " " + pers.getApellidoPaterno() +
", ya tiene asignado un usuario");
           }

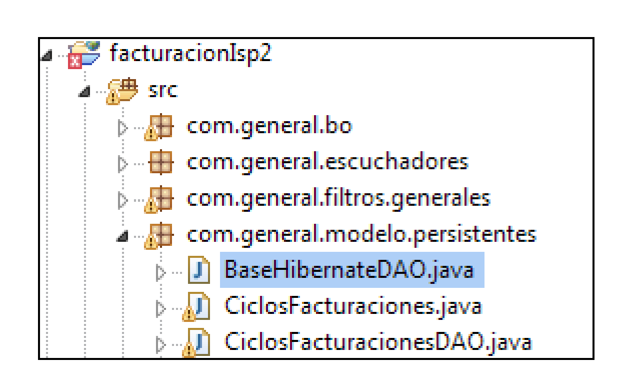

        usuario.setDescUsuario(ingresoUsuarioForm.getDescrip
cionUsuario().toUpperCase());
           usuario.setEstado("A");
           usuario.setFechaRegistro(new Date());

        usuario.setObservacion(ingresoUsuarioForm.getObserva
```

```
cion().toUpperCase());

        usuario.setUsuarioIngreso(descUsuario.toUpperCase())
;
            usuario.setPersonas(pers);
            usuario.setRoles(rolesDao.findById(new
Integer(ingresoUsuarioForm.getRol())) );
            usuario.setClave(
UtilCryptography.encriptar("12345678"));
            usuarioDao.save(usuario);

        }
}
```




```
package com.general.modelo.persistentes;

/*
 * FECHA: 20/06/2009
 * AUTOR: GRUPO 3
 * COMENTARIO: CLASE QUE SE ENCARGA DE DESPACHAR LAS
SESSIONES DE HIBERNATE,
 * DE ESTA CLASE HEREDAN LAS CLASES DAO DE CADA TABLA, SI
SE BORRA ESTA CLASE
 * SALDRA ERROR EN CADENA DE LOS DAOS DE CADA UNA DE LAS
CLASES PERSISTENTES
 * ESTO LO GENERA EL Persistence Tools DE MY ECLIPSE
 * */

import org.hibernate.Session;

import com.general.util.DAOUtil;
```

```
/**
 * Data access object (DAO) for domain model
 * @author MyEclipse Persistence Tools
 */
public class BaseHibernateDAO implements IBaseHibernateDAO
{

	public Session getSession() {
	    return
DAOUtil.getSessionFactory().getCurrentSession();
	}

}
```

com.general.modelo.persistentes
BaseHibernateDAO.java
CiclosFacturaciones.java
CiclosFacturacionesDAO.java
Clientes.java
ClientesDAO.java

```
package com.general.modelo.persistentes;
/*
 * FECHA: 20/06/2009
 * AUTOR: GRUPO 3
 * COMENTARIO: CLASE BEAN QUE REPRESENTA LA TABLA
ciclos_facturaciones DE LA BASE DE DATOS
 *
 * */
import java.util.Date;
import java.util.HashSet;
import java.util.Set;

/**
 * CiclosFacturaciones entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class CiclosFacturaciones implements
java.io.Serializable {

	// Fields

	private Integer idCiclo;
	private String descripcionCiclo;
```

```
        private String estado;
        private Date fechaRegistro;
        private String observacion;
        private Date fechaActualizacion;
        private String usuarioIngreso;
        private Set contratoses = new HashSet(0);
    private Integer diaInicio;
    private Integer diaFin;
        // Constructors

        public Integer getDiaFin() {
           return diaFin;
        }

        public void setDiaFin(Integer diaFin) {
           this.diaFin = diaFin;
        }

        public Integer getDiaInicio() {
           return diaInicio;
        }

        public void setDiaInicio(Integer diaInicio) {
           this.diaInicio = diaInicio;
        }

        /** default constructor */
        public CiclosFacturaciones() {
        }

        /** minimal constructor */
        public CiclosFacturaciones(String descripcionCiclo,
String estado,
                Date fechaRegistro) {
           this.descripcionCiclo = descripcionCiclo;
           this.estado = estado;
           this.fechaRegistro = fechaRegistro;
        }

        /** full constructor */
        public CiclosFacturaciones(String descripcionCiclo,
String estado,
                Date  fechaRegistro,  String  observacion,
Date fechaActualizacion,
                String usuarioIngreso, Set contratoses) {
           this.descripcionCiclo = descripcionCiclo;
           this.estado = estado;
           this.fechaRegistro = fechaRegistro;
           this.observacion = observacion;
```

```
            this.fechaActualizacion = fechaActualizacion;
            this.usuarioIngreso = usuarioIngreso;
            this.contratoses = contratoses;
        }

        // Property accessors

        public Integer getIdCiclo() {
            return this.idCiclo;
        }

        public void setIdCiclo(Integer idCiclo) {
            this.idCiclo = idCiclo;
        }

        public String getDescripcionCiclo() {
            return this.descripcionCiclo;
        }

        public       void       setDescripcionCiclo(String
descripcionCiclo) {
            this.descripcionCiclo = descripcionCiclo;
        }

        public String getEstado() {
            return this.estado;
        }

        public void setEstado(String estado) {
            this.estado = estado;
        }

        public Date getFechaRegistro() {
            return this.fechaRegistro;
        }

        public void setFechaRegistro(Date fechaRegistro) {
            this.fechaRegistro = fechaRegistro;
        }

        public String getObservacion() {
            return this.observacion;
        }

        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

        public Date getFechaActualizacion() {
```

```
            return this.fechaActualizacion;
        }

        public       void       setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }

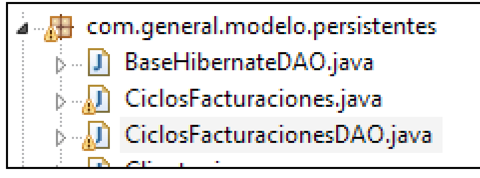
        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public        void        setUsuarioIngreso(String
usuarioIngreso) {
            this.usuarioIngreso = usuarioIngreso;
        }

        public Set getContratoses() {
            return this.contratoses;
        }

        public void setContratoses(Set contratoses) {
            this.contratoses = contratoses;
        }

}
```




```
package com.general.modelo.persistentes;

/*
 * FECHA: 20/06/2009
 * AUTOR: GRUPO 3
 *  COMENTARIO:  CLASE  DAO  QUE  PERMITE  ACCEDER  A  LOS
RTEGITRSO DE LA TABLA CICLOS_FACTURACIONES
 *
 * */

import java.util.Date;
import java.util.List;
import java.util.Set;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.hibernate.LockMode;
```

```
import org.hibernate.Query;
import org.hibernate.criterion.Example;

/**
 * A data access object (DAO) providing persistence and
search support for
 * CiclosFacturaciones entities. Transaction control of the
save(), update() and
 * delete() operations can directly support Spring
container-managed
 * transactions or they can be augmented to handle user-
managed Spring
 * transactions. Each of these methods provides additional
information for how
 * to configure it for the desired type of transaction
control.
 *
 * @see com.general.modelo.persistentes.CiclosFacturaciones
 * @author MyEclipse Persistence Tools
 */

public class CiclosFacturacionesDAO extends
BaseHibernateDAO {
	private static final Log log = LogFactory
			.getLog(CiclosFacturacionesDAO.class);
	// property constants
	public static final String DESCRIPCION_CICLO =
"descripcionCiclo";
	public static final String ESTADO = "estado";
	public static final String OBSERVACION =
"observacion";
	public static final String USUARIO_INGRESO =
"usuarioIngreso";

	public void save(CiclosFacturaciones
transientInstance) {
		log.debug("saving CiclosFacturaciones instance");
		try {
			getSession().save(transientInstance);
			log.debug("save successful");
		} catch (RuntimeException re) {
			log.error("save failed", re);
			throw re;
		}
	}

	public void delete(CiclosFacturaciones
persistentInstance) {
		log.debug("deleting CiclosFacturaciones
```

```
instance");
          try {
               getSession().delete(persistentInstance);
               log.debug("delete successful");
          } catch (RuntimeException re) {
               log.error("delete failed", re);
               throw re;
          }
       }

       public                          CiclosFacturaciones
findById(java.lang.Integer id) {
          log.debug("getting  CiclosFacturaciones  instance
with id: " + id);
          try {
               CiclosFacturaciones       instance       =
(CiclosFacturaciones) getSession()

       .get("com.general.modelo.persistentes.CiclosFacturac
iones",
                                        id);
               return instance;
          } catch (RuntimeException re) {
               log.error("get failed", re);
               throw re;
          }
       }

       public     List     findByExample(CiclosFacturaciones
instance) {
          log.debug("finding  CiclosFacturaciones  instance
by example");
          try {
               List results = getSession().createCriteria(

       "com.general.modelo.persistentes.CiclosFacturaciones
").add(

       Example.create(instance)).list();
               log.debug("find   by   example   successful,
result size: "
                          + results.size());
               return results;
          } catch (RuntimeException re) {
               log.error("find by example failed", re);
               throw re;
          }
       }
```

```
	public List findByProperty(String propertyName, Object value) {
		log.debug("finding CiclosFacturaciones instance with property: "
				+ propertyName + ", value: " + value);
		try {
			String queryString = "from CiclosFacturaciones as model where model."
					+ propertyName + "= ?";
			Query queryObject = getSession().createQuery(queryString);
			queryObject.setParameter(0, value);
			return queryObject.list();
		} catch (RuntimeException re) {
			log.error("find by property name failed", re);
			throw re;
		}
	}

	public List findByDescripcionCiclo(Object descripcionCiclo) {
		return findByProperty(DESCRIPCION_CICLO, descripcionCiclo);
	}

	public List findByEstado(Object estado) {
		return findByProperty(ESTADO, estado);
	}

	public List findByObservacion(Object observacion) {
		return findByProperty(OBSERVACION, observacion);
	}

	public List findByUsuarioIngreso(Object usuarioIngreso) {
		return findByProperty(USUARIO_INGRESO, usuarioIngreso);
	}

	public List findAll() {
		log.debug("finding all CiclosFacturaciones instances");
		try {
			String queryString = "from CiclosFacturaciones";
			Query queryObject = getSession().createQuery(queryString);
			return queryObject.list();
```

```
            } catch (RuntimeException re) {
                  log.error("find all failed", re);
                  throw re;
            }
        }

        public CiclosFacturaciones merge(CiclosFacturaciones
detachedInstance) {
            log.debug("merging            CiclosFacturaciones
instance");
            try {
                  CiclosFacturaciones         result         =
(CiclosFacturaciones) getSession()
                              .merge(detachedInstance);
                  log.debug("merge successful");
                  return result;
            } catch (RuntimeException re) {
                  log.error("merge failed", re);
                  throw re;
            }
        }

        public      void      attachDirty(CiclosFacturaciones
instance) {
            log.debug("attaching   dirty   CiclosFacturaciones
instance");
            try {
                  getSession().saveOrUpdate(instance);
                  log.debug("attach successful");
            } catch (RuntimeException re) {
                  log.error("attach failed", re);
                  throw re;
            }
        }

        public      void      attachClean(CiclosFacturaciones
instance) {
            log.debug("attaching   clean   CiclosFacturaciones
instance");
            try {
                  getSession().lock(instance, LockMode.NONE);
                  log.debug("attach successful");
            } catch (RuntimeException re) {
                  log.error("attach failed", re);
                  throw re;
            }
        }
}
```

com.general.modelo.persistentes
BaseHibernateDAO.java
CiclosFacturaciones.java
CiclosFacturacionesDAO.java
Clientes.java

```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;

/**
 * Clientes entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class Clientes implements java.io.Serializable {

	// Fields

	private Integer idCliente;
	private Personas personas;
	private String direccionEntregaFact;
	private Date fechaInicio;
	private Date fechaRegistro;
	private Date fechaFin;
	private String estado;
	private String telefonoTrabajo;
	private String observacion;
	private Date fechaActualizacion;
	private String usuarioIngreso;
	private Set facturases = new HashSet(0);
	private Set contratoses = new HashSet(0);

	// Constructors

	/** default constructor */
	public Clientes() {
	}

	/** minimal constructor */
	public Clientes(Personas personas, String estado) {
	    this.personas = personas;
	    this.estado = estado;
	}
```

```
        /** full constructor */
        public    Clientes(Personas    personas,    String
direccionEntregaFact,
                Date fechaInicio, Date fechaRegistro, Date
fechaFin, String estado,
                String telefonoTrabajo, String observacion,
                Date      fechaActualizacion,      String
usuarioIngreso, Set facturases,
                Set contratoses) {
            this.personas = personas;
            this.direccionEntregaFact = direccionEntregaFact;
            this.fechaInicio = fechaInicio;
            this.fechaRegistro = fechaRegistro;
            this.fechaFin = fechaFin;
            this.estado = estado;
            this.telefonoTrabajo = telefonoTrabajo;
            this.observacion = observacion;
            this.fechaActualizacion = fechaActualizacion;
            this.usuarioIngreso = usuarioIngreso;
            this.facturases = facturases;
            this.contratoses = contratoses;
        }

        // Property accessors

        public Integer getIdCliente() {
            return this.idCliente;
        }

        public void setIdCliente(Integer idCliente) {
            this.idCliente = idCliente;
        }

        public Personas getPersonas() {
            return this.personas;
        }

        public void setPersonas(Personas personas) {
            this.personas = personas;
        }


        public String getDireccionEntregaFact() {
            return direccionEntregaFact;
        }

        public      void      setDireccionEntregaFact(String
direccionEntregaFact) {
            this.direccionEntregaFact = direccionEntregaFact;
```

```
        }

        public Date getFechaInicio() {
           return this.fechaInicio;
        }

        public void setFechaInicio(Date fechaInicio) {
           this.fechaInicio = fechaInicio;
        }

        public Date getFechaRegistro() {
           return this.fechaRegistro;
        }

        public void setFechaRegistro(Date fechaRegistro) {
           this.fechaRegistro = fechaRegistro;
        }

        public Date getFechaFin() {
           return this.fechaFin;
        }

        public void setFechaFin(Date fechaFin) {
           this.fechaFin = fechaFin;
        }

        public String getEstado() {
           return this.estado;
        }

        public void setEstado(String estado) {
           this.estado = estado;
        }

        public String getTelefonoTrabajo() {
           return this.telefonoTrabajo;
        }

        public        void        setTelefonoTrabajo(String
telefonoTrabajo) {
           this.telefonoTrabajo = telefonoTrabajo;
        }

        public String getObservacion() {
           return this.observacion;
        }

        public void setObservacion(String observacion) {
           this.observacion = observacion;
```

```
	}

	public Date getFechaActualizacion() {
	    return this.fechaActualizacion;
	}

	public         void         setFechaActualizacion(Date
fechaActualizacion) {
	    this.fechaActualizacion = fechaActualizacion;
	}

	public String getUsuarioIngreso() {
	    return this.usuarioIngreso;
	}

	public void setUsuarioIngreso(String usuarioIngreso)
{
	    this.usuarioIngreso = usuarioIngreso;
	}

	public Set getFacturases() {
	    return this.facturases;
	}

	public void setFacturases(Set facturases) {
	    this.facturases = facturases;
	}

	public Set getContratoses() {
	    return this.contratoses;
	}

	public void setContratoses(Set contratoses) {
	    this.contratoses = contratoses;
	}

}
```

com.general.modelo.persistentes
- BaseHibernateDAO.java
- CiclosFacturaciones.java
- CiclosFacturacionesDAO.java
- Clientes.java
- ClientesDAO.java

```
package com.general.modelo.persistentes;

import java.util.List;
```

```
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.hibernate.LockMode;
import org.hibernate.Query;
import org.hibernate.criterion.Example;

/**
 * A data access object (DAO) providing persistence and search support for
 * Clientes entities. Transaction control of the save(), update() and delete()
 * operations can directly support Spring container-managed transactions or they
 * can be augmented to handle user-managed Spring transactions. Each of these
 * methods provides additional information for how to configure it for the
 * desired type of transaction control.
 *
 * @see com.general.modelo.persistentes.Clientes
 * @author MyEclipse Persistence Tools
 */

public class ClientesDAO extends BaseHibernateDAO {
	private static final Log log = LogFactory.getLog(ClientesDAO.class);
	// property constants
	public static final String DIRECCION_ENTREGA_FACT = "direccionEntregaFact";
	public static final String ESTADO = "estado";
	public static final String TELEFONO_TRABAJO = "telefonoTrabajo";
	public static final String OBSERVACION = "observacion";
	public static final String USUARIO_INGRESO = "usuarioIngreso";

	public void save(Clientes transientInstance) {
	    log.debug("saving Clientes instance");
	    try {
	         getSession().save(transientInstance);
	         log.debug("save successful");
	    } catch (RuntimeException re) {
	         log.error("save failed", re);
	         throw re;
	    }
	}

	public void delete(Clientes persistentInstance) {
```

```
            log.debug("deleting Clientes instance");
            try {
                  getSession().delete(persistentInstance);
                  log.debug("delete successful");
            } catch (RuntimeException re) {
                  log.error("delete failed", re);
                  throw re;
            }
        }

        public Clientes findById(java.lang.Integer id) {
            log.debug("getting Clientes instance with id: " +
id);
            try {
                  Clientes      instance      =      (Clientes)
getSession().get(

        "com.general.modelo.persistentes.Clientes", id);
                  return instance;
            } catch (RuntimeException re) {
                  log.error("get failed", re);
                  throw re;
            }
        }

        public List findByExample(Clientes instance) {
            log.debug("finding    Clientes    instance    by
example");
            try {
                  List results = getSession().createCriteria(

        "com.general.modelo.persistentes.Clientes").add(
                              Example.create(instance)).list();
                  log.debug("find   by   example   successful,
result size: "
                              + results.size());
                  return results;
            } catch (RuntimeException re) {
                  log.error("find by example failed", re);
                  throw re;
            }
        }

        public   List   findByProperty(String   propertyName,
Object value) {
            log.debug("finding   Clientes   instance   with
property: " + propertyName
                        + ", value: " + value);
            try {
```

```
                String  queryString  =  "from  Clientes  as
model where model."
                           + propertyName + "= ?";
                Query              queryObject              =
getSession().createQuery(queryString);
                queryObject.setParameter(0, value);
                return queryObject.list();
           } catch (RuntimeException re) {
                log.error("find  by  property  name  failed",
re);
                throw re;
           }
        }

        public     List     findByDireccionEntregaFact(Object
direccionEntregaFact) {
           return     findByProperty(DIRECCION_ENTREGA_FACT,
direccionEntregaFact);
        }

        public List findByEstado(Object estado) {
           return findByProperty(ESTADO, estado);
        }

        public       List       findByTelefonoTrabajo(Object
telefonoTrabajo) {
           return           findByProperty(TELEFONO_TRABAJO,
telefonoTrabajo);
        }

        public List findByObservacion(Object observacion) {
           return findByProperty(OBSERVACION, observacion);
        }

        public       List       findByUsuarioIngreso(Object
usuarioIngreso) {
           return           findByProperty(USUARIO_INGRESO,
usuarioIngreso);
        }

        public List findAll() {
           log.debug("finding all Clientes instances");
           try {
                String queryString = "from Clientes";
                Query              queryObject              =
getSession().createQuery(queryString);
                return queryObject.list();
           } catch (RuntimeException re) {
                log.error("find all failed", re);
```

```
                throw re;
            }
        }

        public Clientes merge(Clientes detachedInstance) {
            log.debug("merging Clientes instance");
            try {
                Clientes      result      =      (Clientes)
getSession().merge(detachedInstance);
                log.debug("merge successful");
                return result;
            } catch (RuntimeException re) {
                log.error("merge failed", re);
                throw re;
            }
        }

        public void attachDirty(Clientes instance) {
            log.debug("attaching dirty Clientes instance");
            try {
                getSession().saveOrUpdate(instance);
                log.debug("attach successful");
            } catch (RuntimeException re) {
                log.error("attach failed", re);
                throw re;
            }
        }

        public void attachClean(Clientes instance) {
            log.debug("attaching clean Clientes instance");
            try {
                getSession().lock(instance, LockMode.NONE);
                log.debug("attach successful");
            } catch (RuntimeException re) {
                log.error("attach failed", re);
                throw re;
            }
        }
}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
Contratos.java

```
package com.general.modelo.persistentes;

/*
 * FECHA: 20/06/2009
 * AUTOR: GRUPO 3
 * COMENTARIO: CLASE BEAN QUE REPRESENTA LA TABLA CONTRATOS
DE LA BASE DE DATOS
 *
 * */

import java.util.Date;
import java.util.HashSet;
import java.util.Set;

/**
 * Contratos entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class Contratos implements java.io.Serializable {

	// Fields

	private Integer idContrato;
	private Empleados empleados;
	private CiclosFacturaciones ciclosFacturaciones;
	private Clientes clientes;
	private String cuenta;
	private Date fechaInicio;
	private Date fechaRegistro;
	private String estado;
	private Date fechaFin;
	private Date fechaActualizacion;
	private String plazo;
	private String observacion;
	private String usuarioIngreso;
	private String valorReferencia1;
	private Set facturases = new HashSet(0);
```

```
        private Set serviciosContratadoses = new HashSet(0);
        private Double precioPlan;

        // Constructors

        public Double getPrecioPlan() {
           return precioPlan;
        }

        public void setPrecioPlan(Double precioPlan) {
           this.precioPlan = precioPlan;
        }

        /** default constructor */
        public Contratos() {
        }

        /** minimal constructor */
        public Contratos(Empleados empleados,
                CiclosFacturaciones    ciclosFacturaciones,
Clientes clientes,
                String  cuenta,  Date  fechaInicio,  Date
fechaRegistro, String estado) {
            this.empleados = empleados;
            this.ciclosFacturaciones = ciclosFacturaciones;
            this.clientes = clientes;
            this.cuenta = cuenta;
            this.fechaInicio = fechaInicio;
            this.fechaRegistro = fechaRegistro;
            this.estado = estado;
        }

        /** full constructor */
        public Contratos(Empleados empleados,
                CiclosFacturaciones    ciclosFacturaciones,
Clientes clientes,
                String  cuenta,  Date  fechaInicio,  Date
fechaRegistro, String estado,
                Date  fechaFin,  Date  fechaActualizacion,
String plazo,
                String observacion, String usuarioIngreso,
String valorReferencia1,
                Set facturases, Set serviciosContratadoses)
{
            this.empleados = empleados;
            this.ciclosFacturaciones = ciclosFacturaciones;
            this.clientes = clientes;
            this.cuenta = cuenta;
            this.fechaInicio = fechaInicio;
```

```
            this.fechaRegistro = fechaRegistro;
            this.estado = estado;
            this.fechaFin = fechaFin;
            this.fechaActualizacion = fechaActualizacion;
            this.plazo = plazo;
            this.observacion = observacion;
            this.usuarioIngreso = usuarioIngreso;
            this.valorReferencia1 = valorReferencia1;
            this.facturases = facturases;
            this.serviciosContratadoses                    =
serviciosContratadoses;
        }

        // Property accessors

        public Integer getIdContrato() {
            return this.idContrato;
        }

        public void setIdContrato(Integer idContrato) {
            this.idContrato = idContrato;
        }

        public Empleados getEmpleados() {
            return this.empleados;
        }

        public void setEmpleados(Empleados empleados) {
            this.empleados = empleados;
        }

        public  CiclosFacturaciones  getCiclosFacturaciones()
{
            return this.ciclosFacturaciones;
        }

        public                                          void
setCiclosFacturaciones(CiclosFacturaciones
ciclosFacturaciones) {
            this.ciclosFacturaciones = ciclosFacturaciones;
        }

        public Clientes getClientes() {
            return this.clientes;
        }

        public void setClientes(Clientes clientes) {
            this.clientes = clientes;
        }
```

```
        public String getCuenta() {
            return this.cuenta;
        }

        public void setCuenta(String cuenta) {
            this.cuenta = cuenta;
        }

        public Date getFechaInicio() {
            return this.fechaInicio;
        }

        public void setFechaInicio(Date fechaInicio) {
            this.fechaInicio = fechaInicio;
        }

        public Date getFechaRegistro() {
            return this.fechaRegistro;
        }

        public void setFechaRegistro(Date fechaRegistro) {
            this.fechaRegistro = fechaRegistro;
        }

        public String getEstado() {
            return this.estado;
        }

        public void setEstado(String estado) {
            this.estado = estado;
        }

        public Date getFechaFin() {
            return this.fechaFin;
        }

        public void setFechaFin(Date fechaFin) {
            this.fechaFin = fechaFin;
        }

        public Date getFechaActualizacion() {
            return this.fechaActualizacion;
        }

        public        void        setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }
```

```
        public String getPlazo() {
           return this.plazo;
        }

        public void setPlazo(String plazo) {
           this.plazo = plazo;
        }

        public String getObservacion() {
           return this.observacion;
        }

        public void setObservacion(String observacion) {
           this.observacion = observacion;
        }

        public String getUsuarioIngreso() {
           return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso)
{
           this.usuarioIngreso = usuarioIngreso;
        }

        public String getValorReferencia1() {
           return this.valorReferencia1;
        }

        public        void        setValorReferencia1(String
valorReferencia1) {
           this.valorReferencia1 = valorReferencia1;
        }

        public Set getFacturases() {
           return this.facturases;
        }

        public void setFacturases(Set facturases) {
           this.facturases = facturases;
        }

        public Set getServiciosContratadoses() {
           return this.serviciosContratadoses;
        }

        public       void       setServiciosContratadoses(Set
serviciosContratadoses) {
```

```
            this.serviciosContratadoses                         =
serviciosContratadoses;
        }

}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
ContratosDAO.java

```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.List;
import java.util.Set;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.hibernate.LockMode;
import org.hibernate.Query;
import org.hibernate.criterion.Example;

/**
 * A data access object (DAO) providing persistence and
search support for
 * Contratos entities. Transaction control of the save(),
update() and delete()
 * operations can directly support Spring container-managed
transactions or they
 * can be augmented to handle user-managed Spring
transactions. Each of these
 * methods provides additional information for how to
configure it for the
 * desired type of transaction control.
 *
 * @see com.general.modelo.persistentes.Contratos
 * @author MyEclipse Persistence Tools
 */

public class ContratosDAO extends BaseHibernateDAO {
        private static final Log log =
LogFactory.getLog(ContratosDAO.class);
        // property constants
        public static final String CUENTA = "cuenta";
        public static final String ESTADO = "estado";
```

```
        public static final String PLAZO = "plazo";
        public    static    final    String    OBSERVACION    =
"observacion";
        public   static   final   String   USUARIO_INGRESO   =
"usuarioIngreso";
        public  static  final  String  VALOR_REFERENCIA1  =
"valorReferencia1";

        public void save(Contratos transientInstance) {
            log.debug("saving Contratos instance");
            try {
                getSession().save(transientInstance);
                log.debug("save successful");
            } catch (RuntimeException re) {
                log.error("save failed", re);
                throw re;
            }
        }

        public void delete(Contratos persistentInstance) {
            log.debug("deleting Contratos instance");
            try {
                getSession().delete(persistentInstance);
                log.debug("delete successful");
            } catch (RuntimeException re) {
                log.error("delete failed", re);
                throw re;
            }
        }

        public Contratos findById(java.lang.Integer id) {
            log.debug("getting Contratos instance with id: "
+ id);
            try {
                Contratos     instance     =     (Contratos)
getSession().get(

        "com.general.modelo.persistentes.Contratos", id);
                return instance;
            } catch (RuntimeException re) {
                log.error("get failed", re);
                throw re;
            }
        }

        public List findByExample(Contratos instance) {
            log.debug("finding    Contratos    instance    by
example");
            try {
```

```
                List results = getSession().createCriteria(

        "com.general.modelo.persistentes.Contratos").add(
                            Example.create(instance)).list();
                log.debug("find   by   example   successful,
result size: "
                            + results.size());
                return results;
           } catch (RuntimeException re) {
                log.error("find by example failed", re);
                throw re;
           }
        }

        public   List   findByProperty(String   propertyName,
Object value) {
           log.debug("finding   Contratos   instance   with
property: " + propertyName
                      + ", value: " + value);
           try {
                String  queryString  =  "from  Contratos  as
model where model."
                            + propertyName + "= ?";
                Query            queryObject            =
getSession().createQuery(queryString);
                queryObject.setParameter(0, value);
                return queryObject.list();
           } catch (RuntimeException re) {
                log.error("find  by  property  name  failed",
re);
                throw re;
           }
        }

        public List findByCuenta(Object cuenta) {
           return findByProperty(CUENTA, cuenta);
        }

        public List findByEstado(Object estado) {
           return findByProperty(ESTADO, estado);
        }

        public List findByPlazo(Object plazo) {
           return findByProperty(PLAZO, plazo);
        }

        public List findByObservacion(Object observacion) {
           return findByProperty(OBSERVACION, observacion);
        }
```

```
        public      List      findByUsuarioIngreso(Object
usuarioIngreso) {
            return          findByProperty(USUARIO_INGRESO,
usuarioIngreso);
        }

        public      List      findByValorReferencia1(Object
valorReferencia1) {
            return          findByProperty(VALOR_REFERENCIA1,
valorReferencia1);
        }

        public List findAll() {
            log.debug("finding all Contratos instances");
            try {
                String queryString = "from Contratos";
                Query             queryObject             =
getSession().createQuery(queryString);
                return queryObject.list();
            } catch (RuntimeException re) {
                log.error("find all failed", re);
                throw re;
            }
        }

        public Contratos merge(Contratos detachedInstance) {
            log.debug("merging Contratos instance");
            try {
                Contratos     result     =     (Contratos)
getSession().merge(detachedInstance);
                log.debug("merge successful");
                return result;
            } catch (RuntimeException re) {
                log.error("merge failed", re);
                throw re;
            }
        }

        public void attachDirty(Contratos instance) {
            log.debug("attaching dirty Contratos instance");
            try {
                getSession().saveOrUpdate(instance);
                log.debug("attach successful");
            } catch (RuntimeException re) {
                log.error("attach failed", re);
                throw re;
            }
        }
```

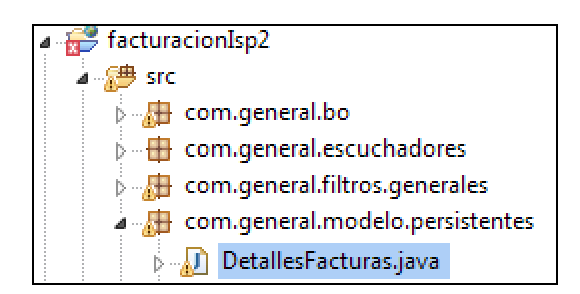
```
        public void attachClean(Contratos instance) {
            log.debug("attaching clean Contratos instance");
            try {
                getSession().lock(instance, LockMode.NONE);
                log.debug("attach successful");
            } catch (RuntimeException re) {
                log.error("attach failed", re);
                throw re;
            }
        }
}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
DatosIp.java

```
package com.general.modelo.persistentes;

public class DatosIp {

        String ip;
        String mac;
        public String getIp() {
            return ip;
        }
        public void setIp(String ip) {
            this.ip = ip;
        }
        public String getMac() {
            return mac;
        }
        public void setMac(String mac) {
            this.mac = mac;
        }
}
```



```
package com.general.modelo.persistentes;

import java.util.Date;

/**
 * DetallesFacturas entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class DetallesFacturas implements java.io.Serializable {

	// Fields

	private Integer idDetalle;
	private Productos productos;
	private Facturas facturas;
	private String descripcion;
	private Double valorUnitario;
	private Integer cantidad;
	private Double subtotal;
	private Double valorIva;
	private Double valorIce;
	private String estado;
	private String observacion;
	private Date fechaRegistro;
	private String usuarioIngreso;
	private Date fechaActualizacion;

	// Constructors

	/** default constructor */
	public DetallesFacturas() {
	}

	/** minimal constructor */
	public DetallesFacturas(Productos productos, Facturas 
facturas,
			Integer cantidad, String estado) {
	    this.productos = productos;
	    this.facturas = facturas;
	    this.cantidad = cantidad;
	    this.estado = estado;
	}

	/** full constructor */
	public DetallesFacturas(Productos productos, Facturas 
facturas,
			String descripcion, Double valorUnitario, 
Integer cantidad,
			Double subtotal, Double valorIva, Double 
valorIce, String estado,
			String observacion, Date fechaRegistro, String 
usuarioIngreso,
```

```
                 Date fechaActualizacion) {
        this.productos = productos;
        this.facturas = facturas;
        this.descripcion = descripcion;
        this.valorUnitario = valorUnitario;
        this.cantidad = cantidad;
        this.subtotal = subtotal;
        this.valorIva = valorIva;
        this.valorIce = valorIce;
        this.estado = estado;
        this.observacion = observacion;
        this.fechaRegistro = fechaRegistro;
        this.usuarioIngreso = usuarioIngreso;
        this.fechaActualizacion = fechaActualizacion;
    }

    // Property accessors

    public Integer getIdDetalle() {
        return this.idDetalle;
    }

    public void setIdDetalle(Integer idDetalle) {
        this.idDetalle = idDetalle;
    }

    public Productos getProductos() {
        return this.productos;
    }

    public void setProductos(Productos productos) {
        this.productos = productos;
    }

    public Facturas getFacturas() {
        return this.facturas;
    }

    public void setFacturas(Facturas facturas) {
        this.facturas = facturas;
    }

    public String getDescripcion() {
        return this.descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public Double getValorUnitario() {
        return this.valorUnitario;
    }
```

```
        public void setValorUnitario(Double valorUnitario) {
            this.valorUnitario = valorUnitario;
        }

        public Integer getCantidad() {
            return this.cantidad;
        }

        public void setCantidad(Integer cantidad) {
            this.cantidad = cantidad;
        }

        public Double getSubtotal() {
            return this.subtotal;
        }

        public void setSubtotal(Double subtotal) {
            this.subtotal = subtotal;
        }

        public Double getValorIva() {
            return this.valorIva;
        }

        public void setValorIva(Double valorIva) {
            this.valorIva = valorIva;
        }

        public Double getValorIce() {
            return this.valorIce;
        }

        public void setValorIce(Double valorIce) {
            this.valorIce = valorIce;
        }

        public String getEstado() {
            return this.estado;
        }

        public void setEstado(String estado) {
            this.estado = estado;
        }

        public String getObservacion() {
            return this.observacion;
        }

        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

        public Date getFechaRegistro() {
            return this.fechaRegistro;
```

```
        }

        public void setFechaRegistro(Date fechaRegistro) {
            this.fechaRegistro = fechaRegistro;
        }

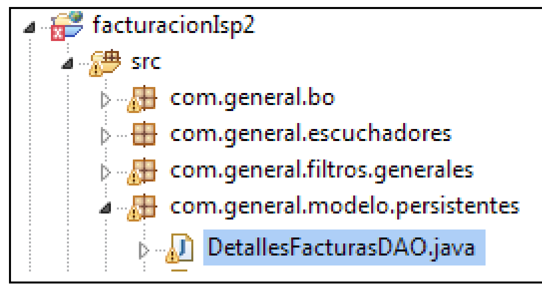
        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso) {
            this.usuarioIngreso = usuarioIngreso;
        }

        public Date getFechaActualizacion() {
            return this.fechaActualizacion;
        }

        public void setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }

}
```




```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.List;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.hibernate.LockMode;
import org.hibernate.Query;
import org.hibernate.criterion.Example;

/**
 * A data access object (DAO) providing persistence and
search support for
 * DetallesFacturas entities. Transaction control of the
save(), update() and
 * delete() operations can directly support Spring
container-managed
```

```
 * transactions or they can be augmented to handle user-
managed Spring
 * transactions. Each of these methods provides additional
information for how
 * to configure it for the desired type of transaction
control.
 *
 * @see com.general.modelo.persistentes.DetallesFacturas
 * @author MyEclipse Persistence Tools
 */

public class DetallesFacturasDAO extends BaseHibernateDAO
{
	private static final Log log =
LogFactory.getLog(DetallesFacturasDAO.class);
	// property constants
	public static final String DESCRIPCION =
"descripcion";
	public static final String VALOR_UNITARIO =
"valorUnitario";
	public static final String CANTIDAD = "cantidad";
	public static final String SUBTOTAL = "subtotal";
	public static final String VALOR_IVA = "valorIva";
	public static final String VALOR_ICE = "valorIce";
	public static final String ESTADO = "estado";
	public static final String OBSERVACION =
"observacion";
	public static final String USUARIO_INGRESO =
"usuarioIngreso";

	public void save(DetallesFacturas
transientInstance) {
		log.debug("saving DetallesFacturas instance");
		try {
			getSession().save(transientInstance);
			log.debug("save successful");
		} catch (RuntimeException re) {
			log.error("save failed", re);
			throw re;
		}
	}

	public void delete(DetallesFacturas
persistentInstance) {
		log.debug("deleting DetallesFacturas
instance");
		try {
			getSession().delete(persistentInstance);
			log.debug("delete successful");
```

```
            } catch (RuntimeException re) {
                  log.error("delete failed", re);
                  throw re;
            }
        }

        public  DetallesFacturas  findById(java.lang.Integer
id) {
            log.debug("getting   DetallesFacturas   instance
with id: " + id);
            try {
                  DetallesFacturas         instance         =
(DetallesFacturas) getSession().get(

        "com.general.modelo.persistentes.DetallesFacturas",
id);
                  return instance;
            } catch (RuntimeException re) {
                  log.error("get failed", re);
                  throw re;
            }
        }

        public      List      findByExample(DetallesFacturas
instance) {
            log.debug("finding DetallesFacturas instance by
example");
            try {
                  List               results                =
getSession().createCriteria(

        "com.general.modelo.persistentes.DetallesFacturas")
.add(

        Example.create(instance)).list();
                  log.debug("find   by   example   successful,
result size: "
                              + results.size());
                  return results;
            } catch (RuntimeException re) {
                  log.error("find by example failed", re);
                  throw re;
            }
        }

        public  List  findByProperty(String  propertyName,
Object value) {
            log.debug("finding   DetallesFacturas   instance
with property: "
```

```
                      +  propertyName  +  ",  value:  "  +
value);
           try {
                String      queryString      =      "from
DetallesFacturas as model where model."
                           + propertyName + "= ?";
                Query           queryObject            =
getSession().createQuery(queryString);
                queryObject.setParameter(0, value);
                return queryObject.list();
           } catch (RuntimeException re) {
                log.error("find by property name failed",
re);
                throw re;
           }
        }

        public List findByDescripcion(Object descripcion) {
           return              findByProperty(DESCRIPCION,
descripcion);
        }

        public       List       findByValorUnitario(Object
valorUnitario) {
           return           findByProperty(VALOR_UNITARIO,
valorUnitario);
        }

        public List findByCantidad(Object cantidad) {
           return findByProperty(CANTIDAD, cantidad);
        }

        public List findBySubtotal(Object subtotal) {
           return findByProperty(SUBTOTAL, subtotal);
        }

        public List findByValorIva(Object valorIva) {
           return findByProperty(VALOR_IVA, valorIva);
        }

        public List findByValorIce(Object valorIce) {
           return findByProperty(VALOR_ICE, valorIce);
        }

        public List findByEstado(Object estado) {
           return findByProperty(ESTADO, estado);
        }

        public List findByObservacion(Object observacion) {
```

```
            return              findByProperty(OBSERVACION,
observacion);
        }

        public      List      findByUsuarioIngreso(Object
usuarioIngreso) {
            return          findByProperty(USUARIO_INGRESO,
usuarioIngreso);
        }

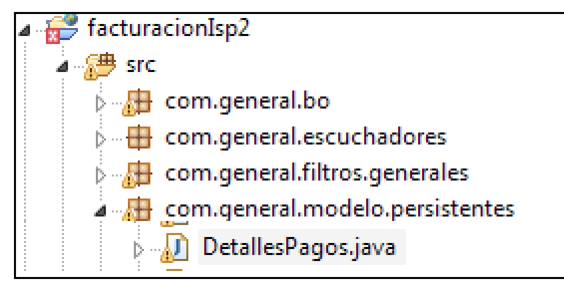
        public List findAll() {
            log.debug("finding     all     DetallesFacturas
instances");
            try {
                 String      queryString      =      "from
DetallesFacturas";
                 Query             queryObject            =
getSession().createQuery(queryString);
                 return queryObject.list();
            } catch (RuntimeException re) {
                 log.error("find all failed", re);
                 throw re;
            }
        }

        public   DetallesFacturas   merge(DetallesFacturas
detachedInstance) {
            log.debug("merging DetallesFacturas instance");
            try {
                 DetallesFacturas         result          =
(DetallesFacturas) getSession().merge(
                           detachedInstance);
                 log.debug("merge successful");
                 return result;
            } catch (RuntimeException re) {
                 log.error("merge failed", re);
                 throw re;
            }
        }

        public  void  attachDirty(DetallesFacturas  instance)
{
            log.debug("attaching   dirty   DetallesFacturas
instance");
            try {
                 getSession().saveOrUpdate(instance);
                 log.debug("attach successful");
            } catch (RuntimeException re) {
                 log.error("attach failed", re);
```

```
                throw re;
            }
        }

        public void attachClean(DetallesFacturas instance)
{
            log.debug("attaching clean DetallesFacturas
instance");
            try {
                getSession().lock(instance,
LockMode.NONE);
                log.debug("attach successful");
            } catch (RuntimeException re) {
                log.error("attach failed", re);
                throw re;
            }
        }
}
```




```
package com.general.modelo.persistentes;

import java.util.Date;
public class DetallesPagos implements java.io.Serializable {

        // Fields

        private Integer idDetallePago;
        private FormasPagos formasPagos;
        private Facturas facturas;
        private Double valor;
        private String descripcion;
        private String referenciaBancaria;
        private String numeroTransaccion;
        private String estado;
        private Date fechaRegistro;
        private Date fechaPago;
        private Date fechaActualizacion;
        private String observacion;
        private String usuarioIngreso;

        // Constructors
```

```
        /** default constructor */
        public DetallesPagos() {
        }

        /** minimal constructor */
        public DetallesPagos(FormasPagos formasPagos, Facturas
facturas,
                String estado) {
            this.formasPagos = formasPagos;
            this.facturas = facturas;
            this.estado = estado;
        }

        /** full constructor */
        public DetallesPagos(FormasPagos formasPagos, Facturas
facturas,
                Double valor, String descripcion, String
referenciaBancaria,
                String numeroTransaccion, String estado, Date
fechaRegistro,
                Date fechaPago, Date fechaActualizacion,
String observacion,
                String usuarioIngreso) {
            this.formasPagos = formasPagos;
            this.facturas = facturas;
            this.valor = valor;
            this.descripcion = descripcion;
            this.referenciaBancaria = referenciaBancaria;
            this.numeroTransaccion = numeroTransaccion;
            this.estado = estado;
            this.fechaRegistro = fechaRegistro;
            this.fechaPago = fechaPago;
            this.fechaActualizacion = fechaActualizacion;
            this.observacion = observacion;
            this.usuarioIngreso = usuarioIngreso;
        }

        // Property accessors

        public Integer getIdDetallePago() {
            return this.idDetallePago;
        }

        public void setIdDetallePago(Integer idDetallePago) {
            this.idDetallePago = idDetallePago;
        }

        public FormasPagos getFormasPagos() {
            return this.formasPagos;
        }

        public void setFormasPagos(FormasPagos formasPagos) {
            this.formasPagos = formasPagos;
        }
```

```
        public Facturas getFacturas() {
            return this.facturas;
        }

        public void setFacturas(Facturas facturas) {
            this.facturas = facturas;
        }

        public Double getValor() {
            return this.valor;
        }

        public void setValor(Double valor) {
            this.valor = valor;
        }

        public String getDescripcion() {
            return this.descripcion;
        }

        public void setDescripcion(String descripcion) {
            this.descripcion = descripcion;
        }

        public String getReferenciaBancaria() {
            return this.referenciaBancaria;
        }

        public void setReferenciaBancaria(String
referenciaBancaria) {
            this.referenciaBancaria = referenciaBancaria;
        }

        public String getNumeroTransaccion() {
            return this.numeroTransaccion;
        }

        public void setNumeroTransaccion(String
numeroTransaccion) {
            this.numeroTransaccion = numeroTransaccion;
        }

        public String getEstado() {
            return this.estado;
        }

        public void setEstado(String estado) {
            this.estado = estado;
        }

        public Date getFechaRegistro() {
            return this.fechaRegistro;
        }
```

```
        public void setFechaRegistro(Date fechaRegistro) {
            this.fechaRegistro = fechaRegistro;
        }

        public Date getFechaPago() {
            return this.fechaPago;
        }

        public void setFechaPago(Date fechaPago) {
            this.fechaPago = fechaPago;
        }

        public Date getFechaActualizacion() {
            return this.fechaActualizacion;
        }

        public void setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }

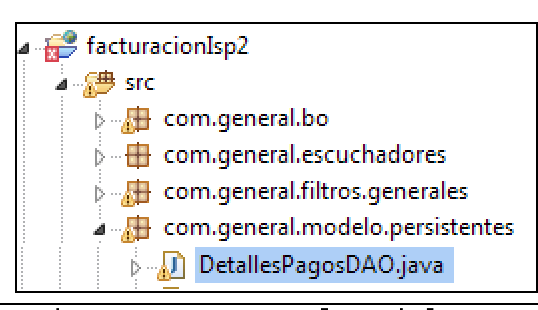
        public String getObservacion() {
            return this.observacion;
        }

        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso) {
            this.usuarioIngreso = usuarioIngreso;
        }

}
```




```
package com.general.modelo.persistentes;
```

```
import java.util.Date;
import java.util.List;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.hibernate.LockMode;
import org.hibernate.Query;
import org.hibernate.criterion.Example;


public class DetallesPagosDAO extends BaseHibernateDAO {
	private static final Log log =
LogFactory.getLog(DetallesPagosDAO.class);
	// property constants
	public static final String VALOR = "valor";
	public static final String DESCRIPCION =
"descripcion";
	public static final String REFERENCIA_BANCARIA =
"referenciaBancaria";
	public static final String NUMERO_TRANSACCION =
"numeroTransaccion";
	public static final String ESTADO = "estado";
	public static final String OBSERVACION =
"observacion";
	public static final String USUARIO_INGRESO =
"usuarioIngreso";

	public void save(DetallesPagos transientInstance) {
	    log.debug("saving DetallesPagos instance");
	    try {
	         getSession().save(transientInstance);
	         log.debug("save successful");
	    } catch (RuntimeException re) {
	         log.error("save failed", re);
	         throw re;
	    }
	}

	public void delete(DetallesPagos persistentInstance)
{
	    log.debug("deleting DetallesPagos instance");
	    try {
	         getSession().delete(persistentInstance);
	         log.debug("delete successful");
	    } catch (RuntimeException re) {
	         log.error("delete failed", re);
	         throw re;
	    }
	}
```

```
        public  DetallesPagos  findById(java.lang.Integer  id)
{
            log.debug("getting  DetallesPagos  instance  with
id: " + id);
            try {
                  DetallesPagos  instance  =  (DetallesPagos)
getSession().get(

        "com.general.modelo.persistentes.DetallesPagos",
id);
                  return instance;
            } catch (RuntimeException re) {
                  log.error("get failed", re);
                  throw re;
            }
        }

        public List findByExample(DetallesPagos instance) {
            log.debug("finding  DetallesPagos  instance  by
example");
            try {
                  List results = getSession().createCriteria(

        "com.general.modelo.persistentes.DetallesPagos").add
(
                              Example.create(instance)).list();
                  log.debug("find  by  example  successful,
result size: "
                              + results.size());
                  return results;
            } catch (RuntimeException re) {
                  log.error("find by example failed", re);
                  throw re;
            }
        }

        public  List  findByProperty(String  propertyName,
Object value) {
            log.debug("finding  DetallesPagos  instance  with
property: "
                        + propertyName + ", value: " + value);
            try {
                  String queryString = "from DetallesPagos as
model where model."
                              + propertyName + "= ?";
                  Query            queryObject            =
getSession().createQuery(queryString);
                  queryObject.setParameter(0, value);
```

```
            return queryObject.list();
        } catch (RuntimeException re) {
            log.error("find  by  property  name  failed",
re);
            throw re;
        }
    }

    public List findByValor(Object valor) {
        return findByProperty(VALOR, valor);
    }

    public List findByDescripcion(Object descripcion) {
        return findByProperty(DESCRIPCION, descripcion);
    }

    public      List      findByReferenciaBancaria(Object
referenciaBancaria) {
        return       findByProperty(REFERENCIA_BANCARIA,
referenciaBancaria);
    }

    public      List      findByNumeroTransaccion(Object
numeroTransaccion) {
        return       findByProperty(NUMERO_TRANSACCION,
numeroTransaccion);
    }

    public List findByEstado(Object estado) {
        return findByProperty(ESTADO, estado);
    }

    public List findByObservacion(Object observacion) {
        return findByProperty(OBSERVACION, observacion);
    }

    public       List       findByUsuarioIngreso(Object
usuarioIngreso) {
        return          findByProperty(USUARIO_INGRESO,
usuarioIngreso);
    }

    public List findAll() {
        log.debug("finding all DetallesPagos instances");
        try {
            String queryString = "from DetallesPagos";
            Query             queryObject              =
getSession().createQuery(queryString);
            return queryObject.list();
```

```
            } catch (RuntimeException re) {
                 log.error("find all failed", re);
                 throw re;
            }
        }

        public      DetallesPagos      merge(DetallesPagos
detachedInstance) {
            log.debug("merging DetallesPagos instance");
            try {
                 DetallesPagos   result   =   (DetallesPagos)
getSession().merge(
                           detachedInstance);
                 log.debug("merge successful");
                 return result;
            } catch (RuntimeException re) {
                 log.error("merge failed", re);
                 throw re;
            }
        }

        public void attachDirty(DetallesPagos instance) {
            log.debug("attaching     dirty     DetallesPagos
instance");
            try {
                 getSession().saveOrUpdate(instance);
                 log.debug("attach successful");
            } catch (RuntimeException re) {
                 log.error("attach failed", re);
                 throw re;
            }
        }

        public void attachClean(DetallesPagos instance) {
            log.debug("attaching     clean     DetallesPagos
instance");
            try {
                 getSession().lock(instance, LockMode.NONE);
                 log.debug("attach successful");
            } catch (RuntimeException re) {
                 log.error("attach failed", re);
                 throw re;
            }
        }
}
```

- facturacionIsp2
  - src
    - com.general.bo
    - com.general.escuchadores
    - com.general.filtros.generales
    - com.general.modelo.persistentes
      - Empleados.java

```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;

/**
 * Empleados entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class Empleados implements java.io.Serializable {

	// Fields

	private Integer idEmpleado;
	private Sucursales sucursales;
	private Personas personas;
	private Empleados empleados;
	private TiposEmpleados tiposEmpleados;
	private Date fechaInicio;
	private Date fechaRegistro;
	private Date fechaFin;
	private Date fechaActualizacion;
	private String estado;
	private String observacion;
	private String usuarioIngreso;
	private Set contratoses = new HashSet(0);
	private Set empleadoses = new HashSet(0);
	private Set facturases = new HashSet(0);

	// Constructors

	/** default constructor */
	public Empleados() {
	}

	/** minimal constructor */
	public  Empleados(Sucursales  sucursales,  Personas
```

```
personas,
                TiposEmpleados    tiposEmpleados,    String
estado) {
          this.sucursales = sucursales;
          this.personas = personas;
          this.tiposEmpleados = tiposEmpleados;
          this.estado = estado;
       }

       /** full constructor */
       public  Empleados(Sucursales  sucursales,  Personas
personas,
                Empleados     empleados,     TiposEmpleados
tiposEmpleados,
                Date fechaInicio, Date fechaRegistro, Date
fechaFin,
                Date  fechaActualizacion,  String  estado,
String observacion,
                String usuarioIngreso, Set contratoses, Set
empleadoses,
                Set facturases) {
          this.sucursales = sucursales;
          this.personas = personas;
          this.empleados = empleados;
          this.tiposEmpleados = tiposEmpleados;
          this.fechaInicio = fechaInicio;
          this.fechaRegistro = fechaRegistro;
          this.fechaFin = fechaFin;
          this.fechaActualizacion = fechaActualizacion;
          this.estado = estado;
          this.observacion = observacion;
          this.usuarioIngreso = usuarioIngreso;
          this.contratoses = contratoses;
          this.empleadoses = empleadoses;
          this.facturases = facturases;
       }

       // Property accessors

       public Integer getIdEmpleado() {
          return this.idEmpleado;
       }

       public void setIdEmpleado(Integer idEmpleado) {
          this.idEmpleado = idEmpleado;
       }

       public Sucursales getSucursales() {
          return this.sucursales;
```

```
        }

        public void setSucursales(Sucursales sucursales) {
           this.sucursales = sucursales;
        }

        public Personas getPersonas() {
           return this.personas;
        }

        public void setPersonas(Personas personas) {
           this.personas = personas;
        }

        public Empleados getEmpleados() {
           return this.empleados;
        }

        public void setEmpleados(Empleados empleados) {
           this.empleados = empleados;
        }

        public TiposEmpleados getTiposEmpleados() {
           return this.tiposEmpleados;
        }

        public     void     setTiposEmpleados(TiposEmpleados
tiposEmpleados) {
           this.tiposEmpleados = tiposEmpleados;
        }

        public Date getFechaInicio() {
           return this.fechaInicio;
        }

        public void setFechaInicio(Date fechaInicio) {
           this.fechaInicio = fechaInicio;
        }

        public Date getFechaRegistro() {
           return this.fechaRegistro;
        }

        public void setFechaRegistro(Date fechaRegistro) {
           this.fechaRegistro = fechaRegistro;
        }

        public Date getFechaFin() {
           return this.fechaFin;
```

```
        }

        public void setFechaFin(Date fechaFin) {
            this.fechaFin = fechaFin;
        }

        public Date getFechaActualizacion() {
            return this.fechaActualizacion;
        }

        public        void        setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }

        public String getEstado() {
            return this.estado;
        }

        public void setEstado(String estado) {
            this.estado = estado;
        }

        public String getObservacion() {
            return this.observacion;
        }

        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso)
{
            this.usuarioIngreso = usuarioIngreso;
        }

        public Set getContratoses() {
            return this.contratoses;
        }

        public void setContratoses(Set contratoses) {
            this.contratoses = contratoses;
        }

        public Set getEmpleadoses() {
```

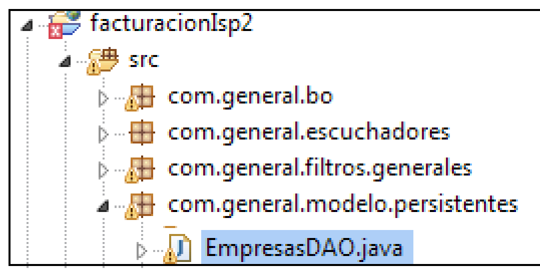
```
            return this.empleadoses;
        }

        public void setEmpleadoses(Set empleadoses) {
            this.empleadoses = empleadoses;
        }

        public Set getFacturases() {
            return this.facturases;
        }

        public void setFacturases(Set facturases) {
            this.facturases = facturases;
        }

}
```




```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.List;
import java.util.Set;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.hibernate.LockMode;
import org.hibernate.Query;
import org.hibernate.criterion.Example;

/**
 * A data access object (DAO) providing persistence and
search support for
 * Empresas entities. Transaction control of the save(),
update() and delete()
 * operations can directly support Spring container-managed
transactions or they
 * can be augmented to handle user-managed Spring
transactions. Each of these
 * methods provides additional information for how to
configure it for the
 * desired type of transaction control.
```

```
 *
 * @see com.general.modelo.persistentes.Empresas
 * @author MyEclipse Persistence Tools
 */

public class EmpresasDAO extends BaseHibernateDAO {
	private static final Log log = LogFactory.getLog(EmpresasDAO.class);
	// property constants
	public static final String RAZON_SOCIAL = "razonSocial";
	public static final String RUC = "ruc";
	public static final String DIRECCION_LOGO = "direccionLogo";
	public static final String ESTADO = "estado";
	public static final String OBSERVACION = "observacion";
	public static final String USUARIO_INGRESO = "usuarioIngreso";

	public void save(Empresas transientInstance) {
	    log.debug("saving Empresas instance");
	    try {
	         getSession().save(transientInstance);
	         log.debug("save successful");
	    } catch (RuntimeException re) {
	         log.error("save failed", re);
	         throw re;
	    }
	}

	public void delete(Empresas persistentInstance) {
	    log.debug("deleting Empresas instance");
	    try {
	         getSession().delete(persistentInstance);
	         log.debug("delete successful");
	    } catch (RuntimeException re) {
	         log.error("delete failed", re);
	         throw re;
	    }
	}

	public Empresas findById(java.lang.Integer id) {
	    log.debug("getting Empresas instance with id: " + id);
	    try {
	         Empresas instance = (Empresas) getSession().get(
```

```
        "com.general.modelo.persistentes.Empresas", id);
                 return instance;
           } catch (RuntimeException re) {
                 log.error("get failed", re);
                 throw re;
           }
        }

        public List findByExample(Empresas instance) {
           log.debug("finding    Empresas    instance    by
example");
           try {
                 List results = getSession().createCriteria(

        "com.general.modelo.persistentes.Empresas").add(
                            Example.create(instance)).list();
                 log.debug("find   by   example   successful,
result size: "
                            + results.size());
                 return results;
           } catch (RuntimeException re) {
                 log.error("find by example failed", re);
                 throw re;
           }
        }

        public   List   findByProperty(String   propertyName,
Object value) {
           log.debug("finding   Empresas   instance   with
property: " + propertyName
                       + ", value: " + value);
           try {
                 String  queryString  =  "from  Empresas  as
model where model."
                            + propertyName + "= ?";
                 Query             queryObject             =
getSession().createQuery(queryString);
                 queryObject.setParameter(0, value);
                 return queryObject.list();
           } catch (RuntimeException re) {
                 log.error("find  by  property  name  failed",
re);
                 throw re;
           }
        }

        public List findByRazonSocial(Object razonSocial) {
           return findByProperty(RAZON_SOCIAL, razonSocial);
        }
```

```
        public List findByRuc(Object ruc) {
            return findByProperty(RUC, ruc);
        }

        public       List       findByDireccionLogo(Object
direccionLogo) {
            return           findByProperty(DIRECCION_LOGO,
direccionLogo);
        }

        public List findByEstado(Object estado) {
            return findByProperty(ESTADO, estado);
        }

        public List findByObservacion(Object observacion) {
            return findByProperty(OBSERVACION, observacion);
        }

        public       List       findByUsuarioIngreso(Object
usuarioIngreso) {
            return           findByProperty(USUARIO_INGRESO,
usuarioIngreso);
        }

        public List findAll() {
            log.debug("finding all Empresas instances");
            try {
                String queryString = "from Empresas";
                Query            queryObject            =
getSession().createQuery(queryString);
                return queryObject.list();
            } catch (RuntimeException re) {
                log.error("find all failed", re);
                throw re;
            }
        }

        public Empresas merge(Empresas detachedInstance) {
            log.debug("merging Empresas instance");
            try {
                Empresas     result     =     (Empresas)
getSession().merge(detachedInstance);
                log.debug("merge successful");
                return result;
            } catch (RuntimeException re) {
                log.error("merge failed", re);
                throw re;
            }
```

```
        }

        public void attachDirty(Empresas instance) {
            log.debug("attaching dirty Empresas instance");
            try {
                  getSession().saveOrUpdate(instance);
                  log.debug("attach successful");
            } catch (RuntimeException re) {
                  log.error("attach failed", re);
                  throw re;
            }
        }

        public void attachClean(Empresas instance) {
            log.debug("attaching clean Empresas instance");
            try {
                  getSession().lock(instance, LockMode.NONE);
                  log.debug("attach successful");
            } catch (RuntimeException re) {
                  log.error("attach failed", re);
                  throw re;
            }
        }
}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
EstadosFacturas.java

```
package com.general.modelo.persistentes;

import java.util.Date;
public class EstadosFacturas implements java.io.Serializable {

        // Fields

        private Integer idEstado;
        private String descripcion;
        private String valor;
        private Date fechaRegistro;
        private Date fechaActualizacion;
        private String observacion;
        private String estado;
        private String usuarioIngreso;
```

```
        // Constructors

        /** default constructor */
        public EstadosFacturas() {
        }

        /** minimal constructor */
        public EstadosFacturas(String estado) {
            this.estado = estado;
        }

        /** full constructor */
        public EstadosFacturas(String descripcion, String valor,
                  Date fechaRegistro, Date fechaActualizacion,
String observacion,
                  String estado, String usuarioIngreso) {
            this.descripcion = descripcion;
            this.valor = valor;
            this.fechaRegistro = fechaRegistro;
            this.fechaActualizacion = fechaActualizacion;
            this.observacion = observacion;
            this.estado = estado;
            this.usuarioIngreso = usuarioIngreso;
        }

        // Property accessors

        public Integer getIdEstado() {
            return this.idEstado;
        }

        public void setIdEstado(Integer idEstado) {
            this.idEstado = idEstado;
        }

        public String getDescripcion() {
            return this.descripcion;
        }

        public void setDescripcion(String descripcion) {
            this.descripcion = descripcion;
        }

        public String getValor() {
            return this.valor;
        }

        public void setValor(String valor) {
            this.valor = valor;
        }

        public Date getFechaRegistro() {
            return this.fechaRegistro;
```

```
        }

        public void setFechaRegistro(Date fechaRegistro) {
            this.fechaRegistro = fechaRegistro;
        }

        public Date getFechaActualizacion() {
            return this.fechaActualizacion;
        }

        public void setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }

        public String getObservacion() {
            return this.observacion;
        }

        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

        public String getEstado() {
            return this.estado;
        }

        public void setEstado(String estado) {
            this.estado = estado;
        }

        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso) {
            this.usuarioIngreso = usuarioIngreso;
        }
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
EstadosFacturasDAO.java

```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.List;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
```

```
import org.hibernate.LockMode;
import org.hibernate.Query;
import org.hibernate.criterion.Example;

/**
 * A data access object (DAO) providing persistence and
search support for
 * EstadosFacturas entities. Transaction control of the
save(), update() and
 * delete() operations can directly support Spring
container-managed
 * transactions or they can be augmented to handle user-
managed Spring
 * transactions. Each of these methods provides additional
information for how
 * to configure it for the desired type of transaction
control.
 *
 * @see com.general.modelo.persistentes.EstadosFacturas
 * @author MyEclipse Persistence Tools
 */

public class EstadosFacturasDAO extends BaseHibernateDAO {
	private static final Log log =
LogFactory.getLog(EstadosFacturasDAO.class);
	// property constants
	public static final String DESCRIPCION =
"descripcion";
	public static final String VALOR = "valor";
	public static final String OBSERVACION =
"observacion";
	public static final String ESTADO = "estado";
	public static final String USUARIO_INGRESO =
"usuarioIngreso";

	public void save(EstadosFacturas transientInstance)
{
		log.debug("saving EstadosFacturas instance");
		try {
			getSession().save(transientInstance);
			log.debug("save successful");
		} catch (RuntimeException re) {
			log.error("save failed", re);
			throw re;
		}
	}

	public void delete(EstadosFacturas
persistentInstance) {
```

```
		log.debug("deleting EstadosFacturas instance");
		try {
			getSession().delete(persistentInstance);
			log.debug("delete successful");
		} catch (RuntimeException re) {
			log.error("delete failed", re);
			throw re;
		}
	}

	public   EstadosFacturas   findById(java.lang.Integer
id) {
		log.debug("getting  EstadosFacturas  instance  with
id: " + id);
		try {
			EstadosFacturas instance = (EstadosFacturas)
getSession().get(

	"com.general.modelo.persistentes.EstadosFacturas",
id);
			return instance;
		} catch (RuntimeException re) {
			log.error("get failed", re);
			throw re;
		}
	}

	public  List  findByExample(EstadosFacturas  instance)
{
		log.debug("finding  EstadosFacturas  instance  by
example");
		try {
			List results = getSession().createCriteria(

	"com.general.modelo.persistentes.EstadosFacturas").a
dd(
					Example.create(instance)).list();
			log.debug("find   by   example   successful,
result size: "
					+ results.size());
			return results;
		} catch (RuntimeException re) {
			log.error("find by example failed", re);
			throw re;
		}
	}

	public   List   findByProperty(String   propertyName,
Object value) {
```

```
            log.debug("finding EstadosFacturas instance with
property: "
                        + propertyName + ", value: " + value);
            try {
                  String queryString = "from EstadosFacturas
as model where model."
                              + propertyName + "= ?";
                  Query              queryObject              =
getSession().createQuery(queryString);
                  queryObject.setParameter(0, value);
                  return queryObject.list();
            } catch (RuntimeException re) {
                  log.error("find by property name failed",
re);
                  throw re;
            }
        }

        public List findByDescripcion(Object descripcion) {
            return findByProperty(DESCRIPCION, descripcion);
        }

        public List findByValor(Object valor) {
            return findByProperty(VALOR, valor);
        }

        public List findByObservacion(Object observacion) {
            return findByProperty(OBSERVACION, observacion);
        }

        public List findByEstado(Object estado) {
            return findByProperty(ESTADO, estado);
        }

        public       List       findByUsuarioIngreso(Object
usuarioIngreso) {
            return           findByProperty(USUARIO_INGRESO,
usuarioIngreso);
        }

        public List findAll() {
            log.debug("finding      all      EstadosFacturas
instances");
            try {
                  String queryString = "from EstadosFacturas";
                  Query              queryObject              =
getSession().createQuery(queryString);
                  return queryObject.list();
            } catch (RuntimeException re) {
```

```
                log.error("find all failed", re);
                throw re;
            }
        }

        public     EstadosFacturas     merge(EstadosFacturas
detachedInstance) {
            log.debug("merging EstadosFacturas instance");
            try {
                EstadosFacturas  result  =  (EstadosFacturas)
getSession().merge(
                            detachedInstance);
                log.debug("merge successful");
                return result;
            } catch (RuntimeException re) {
                log.error("merge failed", re);
                throw re;
            }
        }

        public void attachDirty(EstadosFacturas instance) {
            log.debug("attaching    dirty    EstadosFacturas
instance");
            try {
                getSession().saveOrUpdate(instance);
                log.debug("attach successful");
            } catch (RuntimeException re) {
                log.error("attach failed", re);
                throw re;
            }
        }

        public void attachClean(EstadosFacturas instance) {
            log.debug("attaching    clean    EstadosFacturas
instance");
            try {
                getSession().lock(instance, LockMode.NONE);
                log.debug("attach successful");
            } catch (RuntimeException re) {
                log.error("attach failed", re);
                throw re;
            }
        }
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
Facturas.java

```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;

/**
 * Facturas entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class Facturas implements java.io.Serializable {

        // Fields

        private Integer idFactura;
        private Empleados empleados;
        private Contratos contratos;
        private Clientes clientes;
        private String numeroFactura;
        private Date fechaRegistro;
        private Date fechaFactura;
        private Double subTotal;
        private Double totalIva;
        private Double totalIce;
        private Integer totalItem;
        private Double totalDescuento;
        private String estado;
        private Date fechaActualizacion;
        private String observacion;
        private String usuarioIngreso;
        private Set detallesPagoses = new HashSet(0);
        private Set detallesFacturases = new HashSet(0);

        // Constructors

        /** default constructor */
        public Facturas() {
```

```
        }

        /** minimal constructor */
        public    Facturas(Empleados    empleados,    Clientes
clientes,
                String numeroFactura, String estado) {
           this.empleados = empleados;
           this.clientes = clientes;
           this.numeroFactura = numeroFactura;
           this.estado = estado;
        }

        /** full constructor */
        public   Facturas(Empleados   empleados,   Contratos
contratos,
                Clientes  clientes,  String  numeroFactura,
Date fechaRegistro,
                Date fechaFactura, Double subTotal, Double
totalIva,
                Double totalIce, Integer totalItem, Double
totalDescuento,
                String   estado,   Date   fechaActualizacion,
String observacion,
                String usuarioIngreso, Set detallesPagoses,
Set detallesFacturases) {
           this.empleados = empleados;
           this.contratos = contratos;
           this.clientes = clientes;
           this.numeroFactura = numeroFactura;
           this.fechaRegistro = fechaRegistro;
           this.fechaFactura = fechaFactura;
           this.subTotal = subTotal;
           this.totalIva = totalIva;
           this.totalIce = totalIce;
           this.totalItem = totalItem;
           this.totalDescuento = totalDescuento;
           this.estado = estado;
           this.fechaActualizacion = fechaActualizacion;
           this.observacion = observacion;
           this.usuarioIngreso = usuarioIngreso;
           this.detallesPagoses = detallesPagoses;
           this.detallesFacturases = detallesFacturases;
        }

        // Property accessors

        public Integer getIdFactura() {
           return this.idFactura;
        }
```

```
    public void setIdFactura(Integer idFactura) {
        this.idFactura = idFactura;
    }

    public Empleados getEmpleados() {
        return this.empleados;
    }

    public void setEmpleados(Empleados empleados) {
        this.empleados = empleados;
    }

    public Contratos getContratos() {
        return this.contratos;
    }

    public void setContratos(Contratos contratos) {
        this.contratos = contratos;
    }

    public Clientes getClientes() {
        return this.clientes;
    }

    public void setClientes(Clientes clientes) {
        this.clientes = clientes;
    }

    public String getNumeroFactura() {
        return this.numeroFactura;
    }

    public void setNumeroFactura(String numeroFactura) {
        this.numeroFactura = numeroFactura;
    }

    public Date getFechaRegistro() {
        return this.fechaRegistro;
    }

    public void setFechaRegistro(Date fechaRegistro) {
        this.fechaRegistro = fechaRegistro;
    }

    public Date getFechaFactura() {
        return this.fechaFactura;
    }
```

```
        public void setFechaFactura(Date fechaFactura) {
           this.fechaFactura = fechaFactura;
        }

        public Double getSubTotal() {
           return this.subTotal;
        }

        public void setSubTotal(Double subTotal) {
           this.subTotal = subTotal;
        }

        public Double getTotalIva() {
           return this.totalIva;
        }

        public void setTotalIva(Double totalIva) {
           this.totalIva = totalIva;
        }

        public Double getTotalIce() {
           return this.totalIce;
        }

        public void setTotalIce(Double totalIce) {
           this.totalIce = totalIce;
        }

        public Integer getTotalItem() {
           return this.totalItem;
        }

        public void setTotalItem(Integer totalItem) {
           this.totalItem = totalItem;
        }

        public Double getTotalDescuento() {
           return this.totalDescuento;
        }

        public void setTotalDescuento(Double totalDescuento)
{
           this.totalDescuento = totalDescuento;
        }

        public String getEstado() {
           return this.estado;
        }
```

```
        public void setEstado(String estado) {
            this.estado = estado;
        }

        public Date getFechaActualizacion() {
            return this.fechaActualizacion;
        }

        public        void        setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }

        public String getObservacion() {
            return this.observacion;
        }

        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso)
{
            this.usuarioIngreso = usuarioIngreso;
        }

        public Set getDetallesPagoses() {
            return this.detallesPagoses;
        }

        public void setDetallesPagoses(Set detallesPagoses)
{
            this.detallesPagoses = detallesPagoses;
        }

        public Set getDetallesFacturases() {
            return this.detallesFacturases;
        }

        public        void        setDetallesFacturases(Set
detallesFacturases) {
            this.detallesFacturases = detallesFacturases;
        }

}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
TiposProdcutos.java

```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;

/**
 * TiposProdcutos entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class TiposProdcutos implements java.io.Serializable
{

       // Fields

       private Integer idTipoProducto;
       private String descripcionTipo;
       private String estado;
       private String observacion;
       private Date fechaRegistro;
       private Date fechaActualizacion;
       private String usuarioIngreso;
       private Set productoses = new HashSet(0);
    private Integer caontidadMaxima;
       // Constructors

       public Integer getCaontidadMaxima() {
           return caontidadMaxima;
       }

       public        void        setCaontidadMaxima(Integer
caontidadMaxima) {
           this.caontidadMaxima = caontidadMaxima;
       }

       /** default constructor */
       public TiposProdcutos() {
       }
```

```
        /** minimal constructor */
        public TiposProdcutos(String descripcionTipo, String
estado) {
            this.descripcionTipo = descripcionTipo;
            this.estado = estado;
        }

        /** full constructor */
        public TiposProdcutos(String descripcionTipo, String
estado,
                String   observacion,   Date   fechaRegistro,
Date fechaActualizacion,
                String usuarioIngreso, Set productoses) {
            this.descripcionTipo = descripcionTipo;
            this.estado = estado;
            this.observacion = observacion;
            this.fechaRegistro = fechaRegistro;
            this.fechaActualizacion = fechaActualizacion;
            this.usuarioIngreso = usuarioIngreso;
            this.productoses = productoses;
        }

        // Property accessors

        public Integer getIdTipoProducto() {
            return this.idTipoProducto;
        }

        public        void        setIdTipoProducto(Integer
idTipoProducto) {
            this.idTipoProducto = idTipoProducto;
        }

        public String getDescripcionTipo() {
            return this.descripcionTipo;
        }

        public        void        setDescripcionTipo(String
descripcionTipo) {
            this.descripcionTipo = descripcionTipo;
        }

        public String getEstado() {
            return this.estado;
        }

        public void setEstado(String estado) {
            this.estado = estado;
```

```
        }

        public String getObservacion() {
           return this.observacion;
        }

        public void setObservacion(String observacion) {
           this.observacion = observacion;
        }

        public Date getFechaRegistro() {
           return this.fechaRegistro;
        }

        public void setFechaRegistro(Date fechaRegistro) {
           this.fechaRegistro = fechaRegistro;
        }

        public Date getFechaActualizacion() {
           return this.fechaActualizacion;
        }

        public        void        setFechaActualizacion(Date
fechaActualizacion) {
           this.fechaActualizacion = fechaActualizacion;
        }

        public String getUsuarioIngreso() {
           return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso)
{
           this.usuarioIngreso = usuarioIngreso;
        }

        public Set getProductoses() {
           return this.productoses;
        }

        public void setProductoses(Set productoses) {
           this.productoses = productoses;
        }

}
```

```
facturacionIsp2
  src
    com.general.bo
    com.general.escuchadores
    com.general.filtros.generales
    com.general.modelo.persistentes
      Usuarios.java
```

```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;

/**
 * Usuarios entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class Usuarios implements java.io.Serializable {

	// Fields

	private Integer idUsuario;
	private Roles roles;
	private Personas personas;
	private String descUsuario;
	private String clave;
	private String estado;
	private Date fechaRegistro;
	private String observacion;
	private String usuarioIngreso;
	private Date fechaActualizacion;
	private Set loginUsuarioses = new HashSet(0);
    private String cambioClave;
	// Constructors

	public String getCambioClave() {
	    return cambioClave;
	}

	public void setCambioClave(String cambioClave) {
	    this.cambioClave = cambioClave;
	}

	/** default constructor */
	public Usuarios() {
```

```
        }

        /** minimal constructor */
        public  Usuarios(Roles  roles,  Personas  personas,
String descUsuario,
                 String clave, String estado) {
            this.roles = roles;
            this.personas = personas;
            this.descUsuario = descUsuario;
            this.clave = clave;
            this.estado = estado;
        }

        /** full constructor */
        public  Usuarios(Roles  roles,  Personas  personas,
String descUsuario,
                 String   clave,   String   estado,   Date
fechaRegistro,
                 String  observacion,  String  usuarioIngreso,
Date fechaActualizacion,
                 Set loginUsuarioses) {
            this.roles = roles;
            this.personas = personas;
            this.descUsuario = descUsuario;
            this.clave = clave;
            this.estado = estado;
            this.fechaRegistro = fechaRegistro;
            this.observacion = observacion;
            this.usuarioIngreso = usuarioIngreso;
            this.fechaActualizacion = fechaActualizacion;
            this.loginUsuarioses = loginUsuarioses;
        }

        // Property accessors

        public Integer getIdUsuario() {
            return this.idUsuario;
        }

        public void setIdUsuario(Integer idUsuario) {
            this.idUsuario = idUsuario;
        }

        public Roles getRoles() {
            return this.roles;
        }

        public void setRoles(Roles roles) {
            this.roles = roles;
```

```
    }

    public Personas getPersonas() {
        return this.personas;
    }

    public void setPersonas(Personas personas) {
        this.personas = personas;
    }

    public String getDescUsuario() {
        return this.descUsuario;
    }

    public void setDescUsuario(String descUsuario) {
        this.descUsuario = descUsuario;
    }

    public String getClave() {
        return this.clave;
    }

    public void setClave(String clave) {
        this.clave = clave;
    }

    public String getEstado() {
        return this.estado;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }

    public Date getFechaRegistro() {
        return this.fechaRegistro;
    }

    public void setFechaRegistro(Date fechaRegistro) {
        this.fechaRegistro = fechaRegistro;
    }

    public String getObservacion() {
        return this.observacion;
    }

    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }
```

```
        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso)
{
            this.usuarioIngreso = usuarioIngreso;
        }

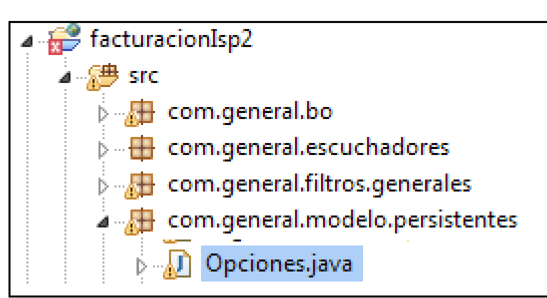
        public Date getFechaActualizacion() {
            return this.fechaActualizacion;
        }

        public        void        setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }

        public Set getLoginUsuarioses() {
            return this.loginUsuarioses;
        }

        public  void  setLoginUsuarioses(Set  loginUsuarioses)
{
            this.loginUsuarioses = loginUsuarioses;
        }

}
```




```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;

/**
```

```
 * Opciones entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class Opciones implements java.io.Serializable {

	// Fields

	private Integer idOpcion;
	private Opciones opciones;
	private String descOpcion;
	private String nombreOpcion;
	private String rutaPagina;
	private String estado;
	private String ayuda;
	private Date fechaRegistro;
	private String observacion;
	private Date fechaActualizacion;
	private String usuarioIngreso;
	private String target;
	private Set opcioneses = new HashSet(0);
	private Set opcionesRoleses = new HashSet(0);
	private Integer ordenPresentacion;
	// Constructors

	public Integer getOrdenPresentacion() {
		return ordenPresentacion;
	}

	public        void        setOrdenPresentacion(Integer
ordenPresentacion) {
		this.ordenPresentacion = ordenPresentacion;
	}

	/** default constructor */
	public Opciones() {
	}

	/** minimal constructor */
	public  Opciones(Integer  idOpcion,  String  descOpcion,
String estado) {
		this.idOpcion = idOpcion;
		this.descOpcion = descOpcion;
		this.estado = estado;
	}

	/** full constructor */
	public  Opciones(Integer  idOpcion,  Opciones  opciones,
```

```
String descOpcion,
                String nombreOpcion, String rutaPagina, String
estado,
                String   ayuda,   Date   fechaRegistro,   String
observacion,
                Date         fechaActualizacion,         String
usuarioIngreso, String target,
                Set opcioneses, Set opcionesRoleses) {
           this.idOpcion = idOpcion;
           this.opciones = opciones;
           this.descOpcion = descOpcion;
           this.nombreOpcion = nombreOpcion;
           this.rutaPagina = rutaPagina;
           this.estado = estado;
           this.ayuda = ayuda;
           this.fechaRegistro = fechaRegistro;
           this.observacion = observacion;
           this.fechaActualizacion = fechaActualizacion;
           this.usuarioIngreso = usuarioIngreso;
           this.target = target;
           this.opcioneses = opcioneses;
           this.opcionesRoleses = opcionesRoleses;
        }

        // Property accessors

        public Integer getIdOpcion() {
           return this.idOpcion;
        }

        public void setIdOpcion(Integer idOpcion) {
           this.idOpcion = idOpcion;
        }

        public Opciones getOpciones() {
           return this.opciones;
        }

        public void setOpciones(Opciones opciones) {
           this.opciones = opciones;
        }

        public String getDescOpcion() {
           return this.descOpcion;
        }

        public void setDescOpcion(String descOpcion) {
           this.descOpcion = descOpcion;
        }
```

```
    public String getNombreOpcion() {
        return this.nombreOpcion;
    }

    public void setNombreOpcion(String nombreOpcion) {
        this.nombreOpcion = nombreOpcion;
    }

    public String getRutaPagina() {
        return this.rutaPagina;
    }

    public void setRutaPagina(String rutaPagina) {
        this.rutaPagina = rutaPagina;
    }

    public String getEstado() {
        return this.estado;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }

    public String getAyuda() {
        return this.ayuda;
    }

    public void setAyuda(String ayuda) {
        this.ayuda = ayuda;
    }

    public Date getFechaRegistro() {
        return this.fechaRegistro;
    }

    public void setFechaRegistro(Date fechaRegistro) {
        this.fechaRegistro = fechaRegistro;
    }

    public String getObservacion() {
        return this.observacion;
    }

    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }
```

```
        public Date getFechaActualizacion() {
            return this.fechaActualizacion;
        }

        public          void          setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }

        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso) {
            this.usuarioIngreso = usuarioIngreso;
        }

        public String getTarget() {
            return this.target;
        }

        public void setTarget(String target) {
            this.target = target;
        }

        public Set getOpcioneses() {
            return this.opcioneses;
        }

        public void setOpcioneses(Set opcioneses) {
            this.opcioneses = opcioneses;
        }

        public Set getOpcionesRoleses() {
            return this.opcionesRoleses;
        }

        public void setOpcionesRoleses(Set opcionesRoleses) {
            this.opcionesRoleses = opcionesRoleses;
        }

}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
OpcionesRoles.java

```
package com.general.modelo.persistentes;

import java.util.Date;

/**
 * OpcionesRoles entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class OpcionesRoles implements java.io.Serializable {

        // Fields

        private Integer idOpcionRol;
        private Roles roles;
        private Opciones opciones;
        private Date fechaRegistro;
        private String estado;
        private String observacion;
        private Date fechaActualizacion;
        private String usuarioIngreso;

        // Constructors

        /** default constructor */
        public OpcionesRoles() {
        }

        /** minimal constructor */
        public OpcionesRoles(Roles roles, Opciones opciones, String
estado) {
            this.roles = roles;
            this.opciones = opciones;
            this.estado = estado;
        }

        /** full constructor */
        public OpcionesRoles(Roles roles, Opciones opciones, Date
fechaRegistro,
                  String estado, String observacion, Date
fechaActualizacion,
                  String usuarioIngreso) {
            this.roles = roles;
```

```
            this.opciones = opciones;
            this.fechaRegistro = fechaRegistro;
            this.estado = estado;
            this.observacion = observacion;
            this.fechaActualizacion = fechaActualizacion;
            this.usuarioIngreso = usuarioIngreso;
        }

        // Property accessors

        public Integer getIdOpcionRol() {
            return this.idOpcionRol;
        }

        public void setIdOpcionRol(Integer idOpcionRol) {
            this.idOpcionRol = idOpcionRol;
        }

        public Roles getRoles() {
            return this.roles;
        }

        public void setRoles(Roles roles) {
            this.roles = roles;
        }

        public Opciones getOpciones() {
            return this.opciones;
        }

        public void setOpciones(Opciones opciones) {
            this.opciones = opciones;
        }

        public Date getFechaRegistro() {
            return this.fechaRegistro;
        }

        public void setFechaRegistro(Date fechaRegistro) {
            this.fechaRegistro = fechaRegistro;
        }

        public String getEstado() {
            return this.estado;
        }

        public void setEstado(String estado) {
            this.estado = estado;
        }

        public String getObservacion() {
            return this.observacion;
        }
```

```
        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

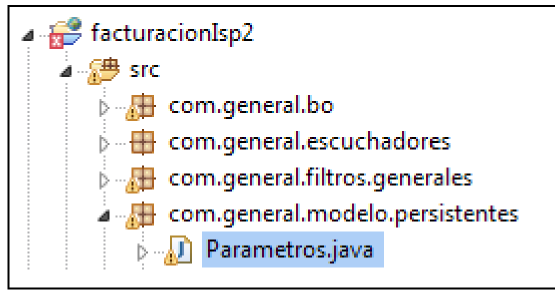
        public Date getFechaActualizacion() {
            return this.fechaActualizacion;
        }

        public void setFechaActualizacion(Date fechaActualizacion)
{
            this.fechaActualizacion = fechaActualizacion;
        }

        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso) {
            this.usuarioIngreso = usuarioIngreso;
        }

}
```




```
package com.general.modelo.persistentes;

import java.util.Date;

/**
 * Parametros generated by MyEclipse Persistence Tools
 */

public class Parametros implements java.io.Serializable {

        // Fields

        private Integer idParametro;

        private TiposParametros tiposParametros;

        private String claveParametro;

        private String valorParametro;
```

```
        private String estado;

        private String descripcion;

        private String observacion;

        private Date fechaRegistro;

        private Date fechaActualizacion;

        private String usuarioIngreso;

        // Constructors

        /** default constructor */
        public Parametros() {
        }

        /** minimal constructor */
        public Parametros(Integer idParametro, TiposParametros
tiposParametros) {
            this.idParametro = idParametro;
            this.tiposParametros = tiposParametros;
        }

        /** full constructor */
        public Parametros(Integer idParametro, TiposParametros
tiposParametros,
                  String claveParametro, String valorParametro,
String estado,
                  String descripcion, String observacion, Date
fechaRegistro,
                  Date fechaActualizacion, String usuarioIngreso) {
            this.idParametro = idParametro;
            this.tiposParametros = tiposParametros;
            this.claveParametro = claveParametro;
            this.valorParametro = valorParametro;
            this.estado = estado;
            this.descripcion = descripcion;
            this.observacion = observacion;
            this.fechaRegistro = fechaRegistro;
            this.fechaActualizacion = fechaActualizacion;
            this.usuarioIngreso = usuarioIngreso;
        }

        // Property accessors

        public Integer getIdParametro() {
            return this.idParametro;
        }

        public void setIdParametro(Integer idParametro) {
            this.idParametro = idParametro;
```

```
        }

        public TiposParametros getTiposParametros() {
            return this.tiposParametros;
        }

        public void setTiposParametros(TiposParametros
tiposParametros) {
            this.tiposParametros = tiposParametros;
        }

        public String getClaveParametro() {
            return this.claveParametro;
        }

        public void setClaveParametro(String claveParametro) {
            this.claveParametro = claveParametro;
        }

        public String getValorParametro() {
            return this.valorParametro;
        }

        public void setValorParametro(String valorParametro) {
            this.valorParametro = valorParametro;
        }

        public String getEstado() {
            return this.estado;
        }

        public void setEstado(String estado) {
            this.estado = estado;
        }

        public String getDescripcion() {
            return this.descripcion;
        }

        public void setDescripcion(String descripcion) {
            this.descripcion = descripcion;
        }

        public String getObservacion() {
            return this.observacion;
        }

        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

        public Date getFechaRegistro() {
            return this.fechaRegistro;
        }
```

```
	public void setFechaRegistro(Date fechaRegistro) {
	    this.fechaRegistro = fechaRegistro;
	}

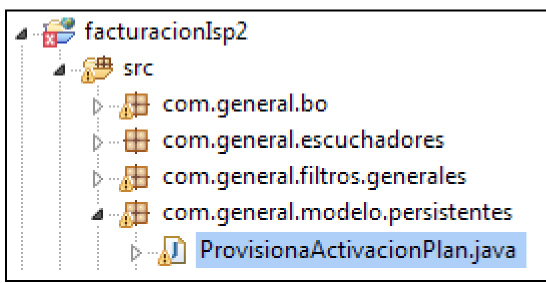
	public Date getFechaActualizacion() {
	    return this.fechaActualizacion;
	}

	public void setFechaActualizacion(Date fechaActualizacion)
{
	    this.fechaActualizacion = fechaActualizacion;
	}

	public String getUsuarioIngreso() {
	    return this.usuarioIngreso;
	}

	public void setUsuarioIngreso(String usuarioIngreso) {
	    this.usuarioIngreso = usuarioIngreso;
	}

}
```




```
package com.general.modelo.persistentes;

import java.util.Date;

/**
 * ProvisionaActivacionPlan generated by MyEclipse Persistence Tools
 */

public class ProvisionaActivacionPlan implements
java.io.Serializable {

	// Fields

	private Integer idAprovisiona;

	private String codiPlan;

	private String valorMaxPlan;

	private String estado;
```

```
        private Date fechaRegistro;

        private Date fechaActualiza;

        private String requerimiento;

        private String respuesta;

        private String usuarioIngreso;

        private String usuarioDespacha;

        private String accion;

        // Constructors

        /** default constructor */
        public ProvisionaActivacionPlan() {
        }

        /** full constructor */
        public ProvisionaActivacionPlan(String codiPlan, String
valorMaxPlan,
                  String estado, Date fechaRegistro, Date
fechaActualiza,
                  String requerimiento, String respuesta, String
usuarioIngreso,
                  String usuarioDespacha, String accion) {
            this.codiPlan = codiPlan;
            this.valorMaxPlan = valorMaxPlan;
            this.estado = estado;
            this.fechaRegistro = fechaRegistro;
            this.fechaActualiza = fechaActualiza;
            this.requerimiento = requerimiento;
            this.respuesta = respuesta;
            this.usuarioIngreso = usuarioIngreso;
            this.usuarioDespacha = usuarioDespacha;
            this.accion = accion;
        }

        // Property accessors

        public Integer getIdAprovisiona() {
            return this.idAprovisiona;
        }

        public void setIdAprovisiona(Integer idAprovisiona) {
            this.idAprovisiona = idAprovisiona;
        }

        public String getCodiPlan() {
            return this.codiPlan;
        }
```

```
public void setCodiPlan(String codiPlan) {
    this.codiPlan = codiPlan;
}

public String getValorMaxPlan() {
    return this.valorMaxPlan;
}

public void setValorMaxPlan(String valorMaxPlan) {
    this.valorMaxPlan = valorMaxPlan;
}

public String getEstado() {
    return this.estado;
}

public void setEstado(String estado) {
    this.estado = estado;
}

public Date getFechaRegistro() {
    return this.fechaRegistro;
}

public void setFechaRegistro(Date fechaRegistro) {
    this.fechaRegistro = fechaRegistro;
}

public Date getFechaActualiza() {
    return this.fechaActualiza;
}

public void setFechaActualiza(Date fechaActualiza) {
    this.fechaActualiza = fechaActualiza;
}

public String getRequerimiento() {
    return this.requerimiento;
}

public void setRequerimiento(String requerimiento) {
    this.requerimiento = requerimiento;
}

public String getRespuesta() {
    return this.respuesta;
}

public void setRespuesta(String respuesta) {
    this.respuesta = respuesta;
}

public String getUsuarioIngreso() {
```

```
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso) {
            this.usuarioIngreso = usuarioIngreso;
        }

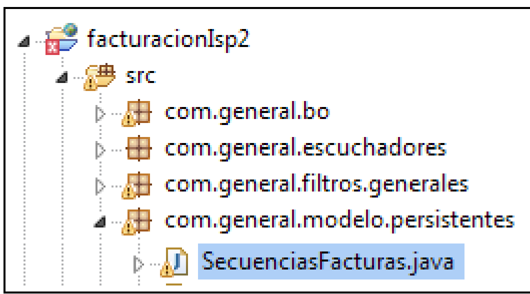
        public String getUsuarioDespacha() {
            return this.usuarioDespacha;
        }

        public void setUsuarioDespacha(String usuarioDespacha) {
            this.usuarioDespacha = usuarioDespacha;
        }

        public String getAccion() {
            return this.accion;
        }

        public void setAccion(String accion) {
            this.accion = accion;
        }

}
```




```
package com.general.modelo.persistentes;

import java.util.Date;

/**
 * SecuenciasFacturas entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class SecuenciasFacturas implements java.io.Serializable {

        // Fields

        private Integer idSecuencia;
        private String numeroInicial;
        private String numeroFinal;
        private String estado;
        private Date fechaRegistro;
```

```
        private Date fechaInicio;
        private Date fechaFin;
        private Date fechaActualizacion;
        private String numeroAutorizacion;
        private String observacion;
        private String usuarioIngreso;
        private Integer siguienteNumero;

        // Constructors

        public Integer getSiguienteNumero() {
            return siguienteNumero;
        }

        public void setSiguienteNumero(Integer siguienteNumero) {
            this.siguienteNumero = siguienteNumero;
        }

        /** default constructor */
        public SecuenciasFacturas() {
        }

        /** full constructor */
        public SecuenciasFacturas(String numeroInicial, String
numeroFinal,
                  String estado, Date fechaRegistro, Date
fechaInicio, Date fechaFin,
                  Date fechaActualizacion, String numeroAutorizacion,
                  String observacion, String usuarioIngreso) {
            this.numeroInicial = numeroInicial;
            this.numeroFinal = numeroFinal;
            this.estado = estado;
            this.fechaRegistro = fechaRegistro;
            this.fechaInicio = fechaInicio;
            this.fechaFin = fechaFin;
            this.fechaActualizacion = fechaActualizacion;
            this.numeroAutorizacion = numeroAutorizacion;
            this.observacion = observacion;
            this.usuarioIngreso = usuarioIngreso;
        }

        // Property accessors

        public Integer getIdSecuencia() {
            return this.idSecuencia;
        }

        public void setIdSecuencia(Integer idSecuencia) {
            this.idSecuencia = idSecuencia;
        }

        public String getNumeroInicial() {
            return this.numeroInicial;
        }
```

```
    public void setNumeroInicial(String numeroInicial) {
        this.numeroInicial = numeroInicial;
    }

    public String getNumeroFinal() {
        return this.numeroFinal;
    }

    public void setNumeroFinal(String numeroFinal) {
        this.numeroFinal = numeroFinal;
    }

    public String getEstado() {
        return this.estado;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }

    public Date getFechaRegistro() {
        return this.fechaRegistro;
    }

    public void setFechaRegistro(Date fechaRegistro) {
        this.fechaRegistro = fechaRegistro;
    }

    public Date getFechaInicio() {
        return this.fechaInicio;
    }

    public void setFechaInicio(Date fechaInicio) {
        this.fechaInicio = fechaInicio;
    }

    public Date getFechaFin() {
        return this.fechaFin;
    }

    public void setFechaFin(Date fechaFin) {
        this.fechaFin = fechaFin;
    }

    public Date getFechaActualizacion() {
        return this.fechaActualizacion;
    }

    public void setFechaActualizacion(Date fechaActualizacion) {
        this.fechaActualizacion = fechaActualizacion;
    }

    public String getNumeroAutorizacion() {
```

```
            return this.numeroAutorizacion;
        }

        public void setNumeroAutorizacion(String numeroAutorizacion)
{
            this.numeroAutorizacion = numeroAutorizacion;
        }

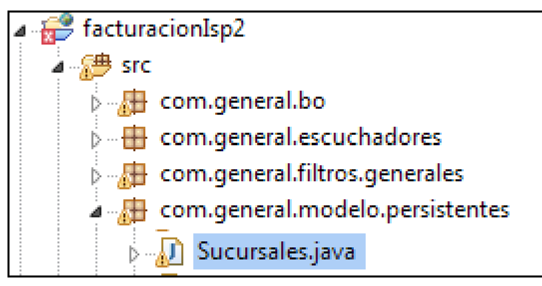
        public String getObservacion() {
            return this.observacion;
        }

        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso) {
            this.usuarioIngreso = usuarioIngreso;
        }

}
```




```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;

/**
 * Sucursales entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class Sucursales implements java.io.Serializable {

        // Fields

        private Integer idSucursal;
```

```
        private Empresas empresas;
        private String nombreSucursal;
        private String direccion;
        private String localidad;
        private Date fechaRegistro;
        private String observacion;
        private String estado;
        private String usuarioIngreso;
        private Date fechaActualizacion;
        private Set empleadoses = new HashSet(0);

        // Constructors

        /** default constructor */
        public Sucursales() {
        }

        /** minimal constructor */
        public    Sucursales(Empresas    empresas,    String
nombreSucursal,
                Date fechaRegistro, String estado) {
            this.empresas = empresas;
            this.nombreSucursal = nombreSucursal;
            this.fechaRegistro = fechaRegistro;
            this.estado = estado;
        }

        /** full constructor */
        public    Sucursales(Empresas    empresas,    String
nombreSucursal,
                String  direccion,  String  localidad,  Date
fechaRegistro,
                String  observacion,  String  estado,  String
usuarioIngreso,
                Date fechaActualizacion, Set empleadoses) {
            this.empresas = empresas;
            this.nombreSucursal = nombreSucursal;
            this.direccion = direccion;
            this.localidad = localidad;
            this.fechaRegistro = fechaRegistro;
            this.observacion = observacion;
            this.estado = estado;
            this.usuarioIngreso = usuarioIngreso;
            this.fechaActualizacion = fechaActualizacion;
            this.empleadoses = empleadoses;
        }

        // Property accessors
```

```
        public Integer getIdSucursal() {
            return this.idSucursal;
        }

        public void setIdSucursal(Integer idSucursal) {
            this.idSucursal = idSucursal;
        }

        public Empresas getEmpresas() {
            return this.empresas;
        }

        public void setEmpresas(Empresas empresas) {
            this.empresas = empresas;
        }

        public String getNombreSucursal() {
            return this.nombreSucursal;
        }

        public  void  setNombreSucursal(String  nombreSucursal)
{
            this.nombreSucursal = nombreSucursal;
        }

        public String getDireccion() {
            return this.direccion;
        }

        public void setDireccion(String direccion) {
            this.direccion = direccion;
        }

        public String getLocalidad() {
            return this.localidad;
        }

        public void setLocalidad(String localidad) {
            this.localidad = localidad;
        }

        public Date getFechaRegistro() {
            return this.fechaRegistro;
        }

        public void setFechaRegistro(Date fechaRegistro) {
            this.fechaRegistro = fechaRegistro;
        }
```

```
        public String getObservacion() {
           return this.observacion;
        }

        public void setObservacion(String observacion) {
           this.observacion = observacion;
        }

        public String getEstado() {
           return this.estado;
        }

        public void setEstado(String estado) {
           this.estado = estado;
        }

        public String getUsuarioIngreso() {
           return this.usuarioIngreso;
        }

        public  void  setUsuarioIngreso(String  usuarioIngreso)
{
           this.usuarioIngreso = usuarioIngreso;
        }

        public Date getFechaActualizacion() {
           return this.fechaActualizacion;
        }

        public         void         setFechaActualizacion(Date
fechaActualizacion) {
           this.fechaActualizacion = fechaActualizacion;
        }

        public Set getEmpleadoses() {
           return this.empleadoses;
        }

        public void setEmpleadoses(Set empleadoses) {
           this.empleadoses = empleadoses;
        }

}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
TiposEmpleados.java

```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;

/**
 * TiposEmpleados entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class TiposEmpleados implements java.io.Serializable {

        // Fields

        private Integer idTipoEmpleado;
        private String descripcionTipo;
        private String estado;
        private Date fechaRegistro;
        private String observacion;
        private Date fechaActualizacion;
        private String usuarioIngreso;
        private Set empleadoses = new HashSet(0);

        // Constructors

        /** default constructor */
        public TiposEmpleados() {
        }

        /** minimal constructor */
        public  TiposEmpleados(String  descripcionTipo,  String
estado,
                 Date fechaRegistro) {
            this.descripcionTipo = descripcionTipo;
            this.estado = estado;
            this.fechaRegistro = fechaRegistro;
        }

        /** full constructor */
```

```
        public  TiposEmpleados(String  descripcionTipo,  String
estado,
                 Date  fechaRegistro,  String  observacion,  Date
fechaActualizacion,
                 String usuarioIngreso, Set empleadoses) {
           this.descripcionTipo = descripcionTipo;
           this.estado = estado;
           this.fechaRegistro = fechaRegistro;
           this.observacion = observacion;
           this.fechaActualizacion = fechaActualizacion;
           this.usuarioIngreso = usuarioIngreso;
           this.empleadoses = empleadoses;
        }

        // Property accessors

        public Integer getIdTipoEmpleado() {
           return this.idTipoEmpleado;
        }

        public  void  setIdTipoEmpleado(Integer  idTipoEmpleado)
{
           this.idTipoEmpleado = idTipoEmpleado;
        }

        public String getDescripcionTipo() {
           return this.descripcionTipo;
        }

        public         void         setDescripcionTipo(String
descripcionTipo) {
           this.descripcionTipo = descripcionTipo;
        }

        public String getEstado() {
           return this.estado;
        }

        public void setEstado(String estado) {
           this.estado = estado;
        }

        public Date getFechaRegistro() {
           return this.fechaRegistro;
        }

        public void setFechaRegistro(Date fechaRegistro) {
           this.fechaRegistro = fechaRegistro;
        }
```

```
        public String getObservacion() {
            return this.observacion;
        }

        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

        public Date getFechaActualizacion() {
            return this.fechaActualizacion;
        }

        public         void         setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }

        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public  void  setUsuarioIngreso(String  usuarioIngreso)
{
            this.usuarioIngreso = usuarioIngreso;
        }

        public Set getEmpleadoses() {
            return this.empleadoses;
        }

        public void setEmpleadoses(Set empleadoses) {
            this.empleadoses = empleadoses;
        }}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
TiposParametros.java

```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;
```

```
/**
 * TiposParametros generated by MyEclipse Persistence Tools
 */

public class TiposParametros implements java.io.Serializable {

	// Fields

	private Integer idTipoParametro;

	private String descripcion;

	private String estado;

	private Date fechaRegistro;

	private String observacion;

	private Date fechaActualizacion;

	private String usuarioIngreso;

	private Set parametroses = new HashSet(0);

	// Constructors

	/** default constructor */
	public TiposParametros() {
	}

	/** minimal constructor */
	public TiposParametros(Integer idTipoParametro, String
estado) {
		this.idTipoParametro = idTipoParametro;
		this.estado = estado;
	}

	/** full constructor */
	public TiposParametros(Integer idTipoParametro, String
descripcion,
			String estado, Date fechaRegistro, String
observacion,
			Date fechaActualizacion, String
usuarioIngreso, Set parametroses) {
		this.idTipoParametro = idTipoParametro;
		this.descripcion = descripcion;
		this.estado = estado;
		this.fechaRegistro = fechaRegistro;
		this.observacion = observacion;
```

```
            this.fechaActualizacion = fechaActualizacion;
            this.usuarioIngreso = usuarioIngreso;
            this.parametroses = parametroses;
        }

        // Property accessors

        public Integer getIdTipoParametro() {
            return this.idTipoParametro;
        }

        public void setIdTipoParametro(Integer idTipoParametro)
{
            this.idTipoParametro = idTipoParametro;
        }

        public String getDescripcion() {
            return this.descripcion;
        }

        public void setDescripcion(String descripcion) {
            this.descripcion = descripcion;
        }

        public String getEstado() {
            return this.estado;
        }

        public void setEstado(String estado) {
            this.estado = estado;
        }

        public Date getFechaRegistro() {
            return this.fechaRegistro;
        }

        public void setFechaRegistro(Date fechaRegistro) {
            this.fechaRegistro = fechaRegistro;
        }

        public String getObservacion() {
            return this.observacion;
        }

        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

        public Date getFechaActualizacion() {
```

```
            return this.fechaActualizacion;
        }

        public          void          setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }

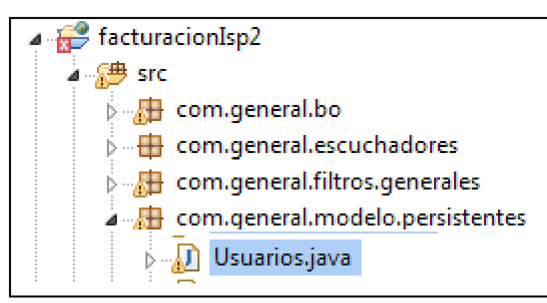
        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso) {
            this.usuarioIngreso = usuarioIngreso;
        }

        public Set getParametroses() {
            return this.parametroses;
        }

        public void setParametroses(Set parametroses) {
            this.parametroses = parametroses;
        }

}
```




```
package com.general.modelo.persistentes;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;

/**
 * Usuarios entity.
 *
 * @author MyEclipse Persistence Tools
 */

public class Usuarios implements java.io.Serializable {
```

```
        // Fields

        private Integer idUsuario;
        private Roles roles;
        private Personas personas;
        private String descUsuario;
        private String clave;
        private String estado;
        private Date fechaRegistro;
        private String observacion;
        private String usuarioIngreso;
        private Date fechaActualizacion;
        private Set loginUsuarioses = new HashSet(0);
    private String cambioClave;
        // Constructors

        public String getCambioClave() {
           return cambioClave;
        }

        public void setCambioClave(String cambioClave) {
           this.cambioClave = cambioClave;
        }

        /** default constructor */
        public Usuarios() {
        }

        /** minimal constructor */
        public  Usuarios(Roles  roles,  Personas  personas,
String descUsuario,
                String clave, String estado) {
            this.roles = roles;
            this.personas = personas;
            this.descUsuario = descUsuario;
            this.clave = clave;
            this.estado = estado;
        }

        /** full constructor */
        public  Usuarios(Roles  roles,  Personas  personas,
String descUsuario,
                String   clave,   String   estado,   Date
fechaRegistro,
                String  observacion,  String  usuarioIngreso,
Date fechaActualizacion,
                Set loginUsuarioses) {
            this.roles = roles;
            this.personas = personas;
```

```
        this.descUsuario = descUsuario;
        this.clave = clave;
        this.estado = estado;
        this.fechaRegistro = fechaRegistro;
        this.observacion = observacion;
        this.usuarioIngreso = usuarioIngreso;
        this.fechaActualizacion = fechaActualizacion;
        this.loginUsuarioses = loginUsuarioses;
    }

    // Property accessors

    public Integer getIdUsuario() {
        return this.idUsuario;
    }

    public void setIdUsuario(Integer idUsuario) {
        this.idUsuario = idUsuario;
    }

    public Roles getRoles() {
        return this.roles;
    }

    public void setRoles(Roles roles) {
        this.roles = roles;
    }

    public Personas getPersonas() {
        return this.personas;
    }

    public void setPersonas(Personas personas) {
        this.personas = personas;
    }

    public String getDescUsuario() {
        return this.descUsuario;
    }

    public void setDescUsuario(String descUsuario) {
        this.descUsuario = descUsuario;
    }

    public String getClave() {
        return this.clave;
    }

    public void setClave(String clave) {
```

```
            this.clave = clave;
        }

        public String getEstado() {
            return this.estado;
        }

        public void setEstado(String estado) {
            this.estado = estado;
        }

        public Date getFechaRegistro() {
            return this.fechaRegistro;
        }

        public void setFechaRegistro(Date fechaRegistro) {
            this.fechaRegistro = fechaRegistro;
        }

        public String getObservacion() {
            return this.observacion;
        }

        public void setObservacion(String observacion) {
            this.observacion = observacion;
        }

        public String getUsuarioIngreso() {
            return this.usuarioIngreso;
        }

        public void setUsuarioIngreso(String usuarioIngreso)
{
            this.usuarioIngreso = usuarioIngreso;
        }

        public Date getFechaActualizacion() {
            return this.fechaActualizacion;
        }

        public void setFechaActualizacion(Date
fechaActualizacion) {
            this.fechaActualizacion = fechaActualizacion;
        }

        public Set getLoginUsuarioses() {
            return this.loginUsuarioses;
        }
```

```
        public void setLoginUsuarioses(Set loginUsuarioses)
{
            this.loginUsuarioses = loginUsuarioses;
        }

}
```

```
facturacionIsp2
  src
    com.general.bo
    com.general.escuchadores
    com.general.filtros.generales
    com.general.modelo.persistentes
    com.general.struts
    com.general.struts.action
      ACambiarClave.java
```

```
package com.general.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import com.general.bo.*;
import com.general.modelo.persistentes.Usuarios;
import com.general.struts.form.FCambioClave;
import com.general.util.DAOUtil;

public class ACambiarClave extends Action
{
        public ActionForward execute(

ActionMapping mapping,

ActionForm form,

HttpServletRequest request,

HttpServletResponse response)throws Exception
       {

FCambioClave l_f=(FCambioClave) form;

try

{
```

```
        LoginBO  l_bo=new LoginBO();

        Usuarios                    user                    =
l_bo.cambiarClave(l_f.getUser(),l_f.getClaveActual(),l_f.getCl
aveNueva());

        request.getSession().setAttribute("USUARIO", user);

        request.getSession().setAttribute("MENSAJEOK", "Cambio
De Clave Realizado Con Exito.");

        return mapping.findForward("okCambiarClave");

}

catch (Exception e)

{

       DAOUtil.rollback();

       e.printStackTrace();

       request.getSession().setAttribute("TITULOERROR",
"Inactivacion De Rol");

       request.getSession().setAttribute("ERRORGENERAL",
e);//new Exception(principalError+e.getMessage()));

       return mapping.getInputForward();

}
       }
}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
com.general.struts
com.general.struts.action
ACambiarClave.java
ActivarImpuestoAction.java
ActivarRolAction.java

```
* Generated by MyEclipse Struts
 * Template path: templates/java/JavaClass.vtl
 */
package com.general.struts.action;

import java.util.Date;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.general.modelo.persistentes.Roles;
import com.general.modelo.persistentes.RolesDAO;
import com.general.util.DAOUtil;

/**
 * MyEclipse Struts
 * Creation date: 10-23-2009
 *
 * XDoclet definition:
 * @struts.action validate="true"
 *     @struts.action-forward     name="ok"     path="sdsdsd"
redirect="true"
 *    @struts.action-forward    name="fallo"    path="sdsdsd"
redirect="true"
 */
public class ActivarRolAction extends Action {
        /*
         * Generated Methods
         */

        /**
         * Method execute
         * @param mapping
         * @param form
         * @param request
         * @param response
         * @return ActionForward
         */
        public  ActionForward  execute(ActionMapping  mapping,
ActionForm form,
                HttpServletRequest                  request,
HttpServletResponse response) {
            // TODO Auto-generated method stub
            try
```

```
            {  //PERSONAENCONTRADA
                 String              idRol               =
request.getParameter("idRol");
                 DAOUtil.beginTransaction();
                 activarRol(idRol);
                 DAOUtil.commit();
                 List rolesFinal = getRolesSistema();

        request.getSession().setAttribute("ROLESSISTEMAS",
rolesFinal);

        request.getSession().setAttribute("MENSAJEOK",      "
Inactivacion Exitosa");
                 return
mapping.findForward("procesoExitoso");
            }
            catch (Exception e)
            {
                 DAOUtil.rollback();
                 e.printStackTrace();

        request.getSession().setAttribute("TITULOERROR",
"Inactivacion De Rol");

        request.getSession().setAttribute("ERRORGENERAL",
e);//new Exception(principalError+e.getMessage()));
                 return
mapping.findForward("procesoFallido");
            }
        }

        private void activarRol(String idRol) {
            // TODO Auto-generated method stub
            RolesDAO rolDao = new RolesDAO();
            Roles rol = rolDao.findById(new Integer(idRol));
            rol.setEstado("A");
            rol.setFechaActualizacion(new Date());
            rolDao.attachDirty(rol);
        }

        public List getRolesSistema ()
        {
            DAOUtil.beginTransaction();
            RolesDAO rolDao = new RolesDAO();
            List rolesSistemas = rolDao.findAll();
            DAOUtil.flush();
            DAOUtil.commit();
            return rolesSistemas;
        }
```

```
}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
com.general.struts
com.general.struts.action
ACambiarClave.java
ActivarImpuestoAction.java
ActivarRolAction.java
ActivarUsuarioAction.java
ActualizarParametrosAction.java

```
* Generated by MyEclipse Struts
 * Template path: templates/java/JavaClass.vtl
 */
package com.general.struts.action;

import java.util.Date;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.general.modelo.persistentes.Parametros;
import com.general.modelo.persistentes.ParametrosDAO;
import com.general.modelo.persistentes.Usuarios;
import com.general.struts.form.ActualizarParametrosForm;
import com.general.util.DAOUtil;
import com.general.util.Sesion;

/**
 * MyEclipse Struts
 * Creation date: 08-02-2007
 *
 * XDoclet definition:
 *          @struts.action          path="/actualizarParametros"
name="actualizarParametrosForm"
input="/paginas/transacciones/seguridad/actualizarParametro.jsp"
scope="request" validate="true"
 *      @struts.action-forward      name="ok"      path="sdfdfdf"
redirect="true"
```

```
 *      @struts.action-forward      name="fallo"      path="dsfsd"
redirect="true"
 */
public class ActualizarParametrosAction extends Action {
	/*
	 * Generated Methods
	 */

	/**
	 * Method execute
	 * @param mapping
	 * @param form
	 * @param request
	 * @param response
	 * @return ActionForward
	 */
	public   ActionForward   execute(ActionMapping   mapping,
ActionForm form,
			HttpServletRequest  request,  HttpServletResponse
response) {
		ActualizarParametrosForm   actualizarParametrosForm   =
(ActualizarParametrosForm)  form;//  TODO  Auto-generated  method
stub
		try
		{
			if
(actualizarParametrosForm.getValorParametro().length() < 1)
			{
				throw new Exception("Debe ingresar un valor
para el parametro");
			}
			DAOUtil.beginTransaction();
			ParametrosDAO parameDao = new ParametrosDAO();
			Parametros    param    =    parameDao.findById(new
Integer(actualizarParametrosForm.getIdParametro()));

	param.setValorParametro(actualizarParametrosForm.getValor
Parametro().toUpperCase());

			if    (param.getDescripcion().equals("IVA")    ||
param.getDescripcion().equals("ICE") )
			{
				try
				{

	Float.parseFloat(actualizarParametrosForm.getValorParamet
ro());

				}catch (Exception e)
```

```
                      {
                           throw  new  Exception("El  valor  del
parametro debe ser numerico, para el impuesto de facturas");
                      }
                }

                //param.setObservacion("Actualizacion         de
parametro");

        param.setUsuarioIngreso(((Usuarios)request.getSession().g
etAttribute(Sesion.USUARIO)).getDescUsuario());
                param.setFechaActualizacion(new Date());
                parameDao.attachDirty(param);
                DAOUtil.commit();
                request.getSession().setAttribute("MENSAJEOK",  "
Actualizacion De Parametro Exitoso");
                request.getSession().setAttribute("PAGINA",
"/filConfigtacion.do");
                return mapping.findForward("ok");
           }catch(Exception e)
           {
                DAOUtil.rollback();
                e.printStackTrace();

        request.getSession().setAttribute("ERRORGENERAL", e);
                request.getSession().setAttribute("TITULOERROR",
"Error Actualizacion Parametro");
                return mapping.findForward("fallo");
           }
        }
}
```

```
facturacionIsp2
  src
    com.general.bo
    com.general.escuchadores
    com.general.filtros.generales
    com.general.modelo.persistentes
    com.general.struts
    com.general.struts.action
      ACambiarClave.java
      ActivarImpuestoAction.java
      ActivarRolAction.java
      ActivarUsuarioAction.java
      ActualizarParametrosAction.java
      ActualizarProductoAction.java
      AniadirOpcionAction.java
```

```
* Generated by MyEclipse Struts
```

```
 * Template path: templates/7java/JavaClass.vtl
 */
package com.general.struts.action;

import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.Set;
import java.util.Vector;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.general.modelo.persistentes.Opciones;
import com.general.modelo.persistentes.OpcionesDAO;
import com.general.modelo.persistentes.OpcionesRoles;
import com.general.modelo.persistentes.OpcionesRolesDAO;
import com.general.modelo.persistentes.Roles;
import com.general.modelo.persistentes.RolesDAO;
import com.general.modelo.persistentes.Usuarios;
import com.general.util.DAOUtil;
import com.general.util.Sesion;

/**
 * MyEclipse Struts
 * Creation date: 08-02-2007
 *
 * XDoclet definition:
 * @struts.action validate="true"
 *                @struts.action-forward                name="ok"
path="/paginas/transacciones/seguridad/gestionOpcionesRoles.jsp"
redirect="true"
 *              @struts.action-forward              name="fallo"
path="/paginas/transacciones/seguridad/gestionOpcionesRoles.jsp"
redirect="true"
 */
public class AniadirOpcionAction extends Action {
	/*
	 * Generated Methods
	 */

	/**
	 * Method execute
	 * @param mapping
	 * @param form
```

```
         * @param request
         * @param response
         * @return ActionForward
         */
        public   ActionForward   execute(ActionMapping   mapping,
ActionForm form,
                HttpServletRequest  request,  HttpServletResponse
response) {
            // TODO Auto-generated method stub
            try
            {
                Roles                    rol                    =
(Roles)request.getSession().getAttribute("ROL_ENC");
                String                 idOpcion                 =
request.getParameter("idOpcion");
                DAOUtil.beginTransaction();
                RolesDAO rolDao = new RolesDAO();
                Roles rol1 = rolDao.findById(rol.getIdRol());
                OpcionesDAO opcDao = new OpcionesDAO();
                Opciones      opc      =      opcDao.findById(new
Integer(idOpcion));

                Set opcionesROl = rol1.getOpcionesRoleses();
                Iterator ite = opcionesROl.iterator();
                OpcionesRoles opsRolFinal = null;
                OpcionesRolesDAO ocoDao = new OpcionesRolesDAO();
                while (ite.hasNext())
                {
                     OpcionesRoles   opsRol   =   (OpcionesRoles)
ite.next();
                     if
(opsRol.getOpciones().getIdOpcion().toString().equals(opc.getIdOp
cion().toString()))
                     {
                          opsRolFinal = opsRol;
                          break;
                     }
                }
                if (opsRolFinal != null )
                {
                     //actualizar con estado A;
                     opsRolFinal.setFechaActualizacion(new
Date());
                     opsRolFinal.setObservacion("Activacion    De
Opcion Desde La Aplicacion");

        opsRolFinal.setUsuarioIngreso(((Usuarios)request.getSessi
on().getAttribute(Sesion.USUARIO)).getDescUsuario());
                     opsRolFinal.setEstado("A");
```

```
                        ocoDao.attachDirty(opsRolFinal);
                }
                else
                {//crear un registro
                        opsRolFinal = new OpcionesRoles();
                        opsRolFinal.setEstado("A");
                        opsRolFinal.setFechaRegistro(new Date());
                        opsRolFinal.setObservacion("Creacion   Desde
La Aplicacion");
                        opsRolFinal.setOpciones(opc);
                        opsRolFinal.setRoles(rol1);

        opsRolFinal.setUsuarioIngreso(((Usuarios)request.getSessi
on().getAttribute(Sesion.USUARIO)).getDescUsuario());
                        ocoDao.save(opsRolFinal);
                }
                //---------------------
                rol1 = rolDao.findById(rol.getIdRol());
//              /obtener los modulos padres del rol
                Vector modulosPadresAsignados = new Vector();
                Set modulosRol = rol1.getOpcionesRoleses();

                Iterator            modulosAsigandos           =
modulosRol.iterator();
                ///Modulos Asignados
                while (modulosAsigandos.hasNext())
                {
                        OpcionesRoles         opcionesRol         =
(OpcionesRoles)modulosAsigandos.next();
                        if (opcionesRol.getEstado().equals("A"))
                        {
                                Opciones           opcion           =
opcionesRol.getOpciones();
                                if (opcion.getOpciones()==null)
                                {
                                        if
(opcion.getEstado().equals("A"))
                                        {

        modulosPadresAsignados.add(opcion);
                                        }
                                }
                        }
                }
                //Modulos no asigandos
                OpcionesDAO opcionesDAo = new OpcionesDAO();
                Vector modulosNoAsigandos = new Vector();
                List                 opciones                  =
opcionesDAo.findByProperty(opcionesDAo.ESTADO, "A");
```

```
                Iterator modulosNAsig = opciones.iterator();
                int bandera = 0;
                while (modulosNAsig.hasNext())
                {
                     Opciones              opcion1              =
(Opciones)modulosNAsig.next();
                     bandera = 0;

                     if (opcion1.getOpciones()==null)
                     {
                          for      (int      i      =      0;
modulosPadresAsignados.size() > i; i++)
                          {
                               Opciones  opcion2  =  (Opciones)
modulosPadresAsignados.elementAt(i);
                               if                            (
opcion2.getIdOpcion().toString().equals(opcion1.getIdOpcion().toS
tring()) )
                               {
                                    bandera  =  1;//encontro  una
ocurrencia
                                    break;
                               }
                          }
                          if (bandera == 0)
                          {
                               //insertarlo
                               modulosNoAsigandos.add(opcion1);
                          }
                     }

                }
                DAOUtil.commit();
                datos(request, rol);
                /*request.getSession().setAttribute("ROL_ENC",
rol1);
                request.getSession().setAttribute("OPC_ASIG",
modulosPadresAsignados);
                request.getSession().setAttribute("OPC_NO_ASIG",
modulosNoAsigandos);*/

                return mapping.findForward("ok");
           }catch(Exception e)
           {
                e.printStackTrace();
                DAOUtil.rollback();
                request.getSession().setAttribute("TITULOERROR",
"Añadir Opciones");
                request.getSession().setAttribute("ERRORGENERAL",
```

```
e);//new Exception(principalError+e.getMessage()));
                return mapping.findForward("fallo");
           }
        }

        public  void  datos  (HttpServletRequest  request,  Roles
rol1) throws Exception
        {
           String idRol = rol1.getIdRol().toString();
           DAOUtil.beginTransaction();
           RolesDAO rolDao = new RolesDAO();
           Roles rol = rolDao.findById(new Integer(idRol));
           if (rol==null)
           {
                throw  new  Exception("No  se  encontro  el  rol  con
codigo " + idRol);
           }
           ///obtener los modulos padres del rol
           Vector modulosPadresAsignados = new Vector();
           Set modulosRol = rol.getOpcionesRoleses();

           Iterator modulosAsigandos = modulosRol.iterator();
           ///Modulos Asignados
           while (modulosAsigandos.hasNext())
           {
                OpcionesRoles            opcionesRol            =
(OpcionesRoles)modulosAsigandos.next();
                if (opcionesRol.getEstado().equals("A"))
                {

                     Opciones              opcion              =
opcionesRol.getOpciones();
                     if (opcion.getOpciones()==null)
                     {
                          if (opcion.getEstado().equals("A"))
                          {

        modulosPadresAsignados.add(opcion);
                          }
                     }
                }
           }
           //Modulos no asigandos
           OpcionesDAO opcionesDAo = new OpcionesDAO();
           Vector modulosNoAsigandos = new Vector();
           List                     opciones                     =
opcionesDAo.findByProperty(opcionesDAo.ESTADO, "A");
           Iterator modulosNAsig = opciones.iterator();
           int bandera = 0;
```

```
            while (modulosNAsig.hasNext())
            {
                  Opciones opcion1 = (Opciones)modulosNAsig.next();
                  bandera = 0;

                  if (opcion1.getOpciones()==null)
                  {
                        for        (int        i        =        0;
modulosPadresAsignados.size() > i; i++)
                        {
                              Opciones    opcion2    =    (Opciones)
modulosPadresAsignados.elementAt(i);
                              if                                  (
opcion2.getIdOpcion().toString().equals(opcion1.getIdOpcion().toS
tring()) )
                              {
                                    bandera   =   1;//encontro   una
ocurrencia
                                    break;
                              }
                        }
                        if (bandera == 0)
                        {
                              //insertarlo
                              modulosNoAsigandos.add(opcion1);
                        }
                  }

            }
            request.getSession().setAttribute("ROL_ENC", rol);
            request.getSession().setAttribute("OPC_ASIG",
modulosPadresAsignados);
            request.getSession().setAttribute("OPC_NO_ASIG",
modulosNoAsigandos);
            DAOUtil.commit();}}
```

facturacionIsp2
src
com.general.bo
com.general.escuchadores
com.general.filtros.generales
com.general.modelo.persistentes
com.general.struts
com.general.struts.action
com.general.struts.form
ActualizarParametrosForm.java
ActualizarProductoForm.java
CambiarClaveUsuarioForm.java

```
/*
 * Generated by MyEclipse Struts
 * Template path: templates/java/JavaClass.vtl
 */
package com.general.struts.form;

import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;

/**
 * MyEclipse Struts
 * Creation date: 08-02-2007
 *
 * XDoclet definition:
 * @struts.form name="generacionFacturasForm"
 */
public class GeneracionFacturasForm extends ActionForm {
	/*
	 * Generated fields
	 */

	/** fecha property */
	private String fecha;

	/** ciclo property */
	private String ciclo;

	/*
	 * Generated Methods
	 */

	/**
```

```
         * Method validate
         * @param mapping
         * @param request
         * @return ActionErrors
         */
        public ActionErrors validate(ActionMapping mapping,
                HttpServletRequest request) {
            // TODO Auto-generated method stub
            return null;
        }

        /**
         * Method reset
         * @param mapping
         * @param request
         */
        public     void     reset(ActionMapping     mapping,
HttpServletRequest request) {
            // TODO Auto-generated method stub
        }

        /**
         * Returns the fecha.
         * @return String
         */
        public String getFecha() {
            return fecha;
        }

        /**
         * Set the fecha.
         * @param fecha The fecha to set
         */
        public void setFecha(String fecha) {
            this.fecha = fecha;
        }

        /**
         * Returns the ciclo.
         * @return String
         */
        public String getCiclo() {
            return ciclo;
        }

        /**
         * Set the ciclo.
         * @param ciclo The ciclo to set
         */
```

```
        public void setCiclo(String ciclo) {
            this.ciclo = ciclo;
        }
}
```

```
/*
 * Generated by MyEclipse Struts
 * Template path: templates/java/JavaClass.vtl
 */
package com.general.struts.form;

import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;

/**
 * MyEclipse Struts
 * Creation date: 08-01-2009
 *
 * XDoclet definition:
 * @struts.form name="ingresoPersonasForm"
 */
public class IngresoClientesForm extends ActionForm {
        /*
         * Generated fields
         */

        /** fechaNacimiento property */
        private String fechaNacimiento;

        /** celular property */
        private String celular;

        /** segundoNombre property */
        private String segundoNombre;

        /** sexo property */
        private String tipoPersona;

        /** apellidoPaterno property */
        private String apellidoPaterno;

        /** cedula property */
        private String cedula;

        /** ruc property */
        private String ruc;
```

```
        /** observacion property */
        private String observacion;

        /** primerNombre property */
        private String primerNombre;

        /** apellidoMaterno property */
        private String apellidoMaterno;

        /** email property */
        private String email;

        /** pasaporte property */
        private String pasaporte;

        /** direccion property */
        private String direccion;

        /** telefono property */
        private String telefono;

        private String sexo;

        private String idPersona;

        private String telefonoTrabajo;

        private String correspondencia;
        /*
         * Generated Methods
         */

        public String getCorrespondencia() {
           return correspondencia;
        }

        public        void        setCorrespondencia(String
correspondencia) {
           this.correspondencia = correspondencia;
        }

        public String getTelefonoTrabajo() {
           return telefonoTrabajo;
        }

        public        void        setTelefonoTrabajo(String
telefonoTrabajo) {
           this.telefonoTrabajo = telefonoTrabajo;
```

```
        }

        public String getIdPersona() {
           return idPersona;
        }

        public void setIdPersona(String idPersona) {
           this.idPersona = idPersona;
        }

        public String getSexo() {
           return sexo;
        }

        public void setSexo(String sexo) {
           this.sexo = sexo;
        }

        /**
         * Method validate
         * @param mapping
         * @param request
         * @return ActionErrors
         */
        public ActionErrors validate(ActionMapping mapping,
                HttpServletRequest request) {
           // TODO Auto-generated method stub
           return null;
        }

        /**
         * Method reset
         * @param mapping
         * @param request
         */
        public     void     reset(ActionMapping     mapping,
HttpServletRequest request) {
           // TODO Auto-generated method stub
        }

        /**
         * Returns the fechaNacimiento.
         * @return String
         */
        public String getFechaNacimiento() {
           return fechaNacimiento;
        }

        /**
```

```
         * Set the fechaNacimiento.
         * @param fechaNacimiento The fechaNacimiento to set
         */
        public        void        setFechaNacimiento(String
fechaNacimiento) {
           this.fechaNacimiento = fechaNacimiento;
        }

        /**
         * Returns the celular.
         * @return String
         */
        public String getCelular() {
           return celular;
        }

        /**
         * Set the celular.
         * @param celular The celular to set
         */
        public void setCelular(String celular) {
           this.celular = celular;
        }

        /**
         * Returns the segundoNombre.
         * @return String
         */
        public String getSegundoNombre() {
           return segundoNombre;
        }

        /**
         * Set the segundoNombre.
         * @param segundoNombre The segundoNombre to set
         */
        public void setSegundoNombre(String segundoNombre) {
           this.segundoNombre = segundoNombre;
        }

        /**
         * Returns the apellidoPaterno.
         * @return String
         */
        public String getApellidoPaterno() {
           return apellidoPaterno;
        }

        /**
```

```
         * Set the apellidoPaterno.
         * @param apellidoPaterno The apellidoPaterno to set
         */
        public         void         setApellidoPaterno(String
apellidoPaterno) {
            this.apellidoPaterno = apellidoPaterno;
        }

        /**
         * Returns the cedula.
         * @return String
         */
        public String getCedula() {
            return cedula;
        }

        /**
         * Set the cedula.
         * @param cedula The cedula to set
         */
        public void setCedula(String cedula) {
            this.cedula = cedula;
        }

        /**
         * Returns the ruc.
         * @return String
         */
        public String getRuc() {
            return ruc;
        }

        /**
         * Set the ruc.
         * @param ruc The ruc to set
         */
        public void setRuc(String ruc) {
            this.ruc = ruc;
        }

        /**
         * Returns the observacion.
         * @return String
         */
        public String getObservacion() {
            return observacion;
        }

        /**
```

```
         * Set the observacion.
         * @param observacion The observacion to set
         */
        public void setObservacion(String observacion) {
           this.observacion = observacion;
        }

        /**
         * Returns the primerNombre.
         * @return String
         */
        public String getPrimerNombre() {
           return primerNombre;
        }

        /**
         * Set the primerNombre.
         * @param primerNombre The primerNombre to set
         */
        public void setPrimerNombre(String primerNombre) {
           this.primerNombre = primerNombre;
        }

        /**
         * Returns the apellidoMaterno.
         * @return String
         */
        public String getApellidoMaterno() {
           return apellidoMaterno;
        }

        /**
         * Set the apellidoMaterno.
         * @param apellidoMaterno The apellidoMaterno to set
         */
        public         void         setApellidoMaterno(String
apellidoMaterno) {
           this.apellidoMaterno = apellidoMaterno;
        }

        /**
         * Returns the email.
         * @return String
         */
        public String getEmail() {
           return email;
        }

        /**
```

```
     * Set the email.
     * @param email The email to set
     */
    public void setEmail(String email) {
       this.email = email;
    }

    /**
     * Returns the pasaporte.
     * @return String
     */
    public String getPasaporte() {
       return pasaporte;
    }

    /**
     * Set the pasaporte.
     * @param pasaporte The pasaporte to set
     */
    public void setPasaporte(String pasaporte) {
       this.pasaporte = pasaporte;
    }

    /**
     * Returns the direccion.
     * @return String
     */
    public String getDireccion() {
       return direccion;
    }

    /**
     * Set the direccion.
     * @param direccion The direccion to set
     */
    public void setDireccion(String direccion) {
       this.direccion = direccion;
    }

    /**
     * Returns the telefono.
     * @return String
     */
    public String getTelefono() {
       return telefono;
    }

    /**
     * Set the telefono.
```

```
         * @param telefono The telefono to set
         */
        public void setTelefono(String telefono) {
            this.telefono = telefono;
        }

        public String getTipoPersona() {
            return tipoPersona;
        }

        public void setTipoPersona(String tipoPersona) {
            this.tipoPersona = tipoPersona;}
```

```
com.general.util
    ActionExportJasperReport.java
    ActionMessagesUtil.java
    CargarArchivo.java
    Configuracion.java
```

```
package com.general.util;

import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.spec.AlgorithmParameterSpec;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.KeySpec;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.PBEParameterSpec;

public class UtilCryptography
{
        private static byte[] SALT_BYTES = {
            (byte)0xA9, (byte)0x9B, (byte)0xC8,
            (byte)0x32,  (byte)0x56, (byte)0x35,
            (byte)0xE3, (byte)0x03      };
        private static int ITERATION_COUNT = 19;
        private          final          static          String
CLAVE="REALMSOFT2006DT01984";
```

```
        public  static  String  encriptar(String  str)  throws
InvalidKeyException,                    InvalidKeySpecException,
NoSuchAlgorithmException,                NoSuchPaddingException,
InvalidAlgorithmParameterException,
IllegalBlockSizeException,                   BadPaddingException,
UnsupportedEncodingException
        {
           return encriptar(CLAVE, str);
        }
        public  static  String  encriptar(String  passPhrase,
String       str)       throws       InvalidKeySpecException,
NoSuchAlgorithmException,                NoSuchPaddingException,
InvalidKeyException,      InvalidAlgorithmParameterException,
IllegalBlockSizeException,                   BadPaddingException,
UnsupportedEncodingException
        {
           Cipher ecipher = null;
           Cipher dcipher = null;
           // Crear la key
           KeySpec          keySpec          =          new
PBEKeySpec(passPhrase.toCharArray(),SALT_BYTES,ITERATION_COUN
T);
           SecretKey                key                    =
SecretKeyFactory.getInstance("PBEWithMD5AndDES").generateSecr
et(keySpec);
           ecipher = Cipher.getInstance(key.getAlgorithm());
           dcipher = Cipher.getInstance(key.getAlgorithm());
           // Preparar los parametros para los ciphers
           AlgorithmParameterSpec     paramSpec     =     new
PBEParameterSpec(SALT_BYTES,ITERATION_COUNT);
           // Crear los ciphers
           ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
           dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
           // Encodear la cadena a bytes usando utf-8
           byte[] utf8 = str.getBytes("UTF8");
           // Encriptar
           byte[] enc = ecipher.doFinal(utf8);
           // Encodear bytes a base64 para obtener cadena
           return new sun.misc.BASE64Encoder().encode(enc);
        }
        public  static  String  desencriptar(String  passPhrase,
String       str)       throws       InvalidKeySpecException,
NoSuchAlgorithmException,                NoSuchPaddingException,
InvalidKeyException,      InvalidAlgorithmParameterException,
IOException, IllegalBlockSizeException, BadPaddingException
        {
           Cipher ecipher = null;
           Cipher dcipher = null;
           // Crear la key
```

```
            KeySpec            keySpec            =          new
PBEKeySpec(passPhrase.toCharArray(),SALT_BYTES,
ITERATION_COUNT);
            SecretKey                  key                     =
SecretKeyFactory.getInstance("PBEWithMD5AndDES").generateSecr
et(keySpec);
            ecipher = Cipher.getInstance(key.getAlgorithm());
            dcipher = Cipher.getInstance(key.getAlgorithm());
            // Preparar los parametros para los ciphers
            AlgorithmParameterSpec     paramSpec     =     new
PBEParameterSpec(SALT_BYTES,ITERATION_COUNT);
            // Crear los ciphers
            ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
            dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
            // Decodear base64 y obtener bytes
            byte[]            dec            =            new
sun.misc.BASE64Decoder().decodeBuffer(str);
            // Desencriptar
            byte[] utf8 = dcipher.doFinal(dec);
            // Decodear usando utf-8
            return new String(utf8, "UTF8");
        }
        public  static  String  desencriptar(String  str)  throws
InvalidKeyException,                  InvalidKeySpecException,
NoSuchAlgorithmException,                NoSuchPaddingException,
InvalidAlgorithmParameterException,
IllegalBlockSizeException, BadPaddingException, IOException
        {
            return desencriptar(CLAVE,str);
        }
        public static void main(String arg[]) throws Exception
        {
            //ZtHJnAGuoo8=

        System.out.println(desencriptar("BG91tPU3tGj78djwYxuJg
Q=="));
        }

}
```

```
package com.general.util;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Vector;
```

```
import com.general.modelo.persistentes.DatosIp;

public class UtilLeeDhcp {

        public static Vector leerDhcp() throws Exception
        {
            Vector vec = new Vector();
            String file ="Z:\\dhcpd.leases";
            String texto="";
             FileReader fr = new FileReader(file);
            BufferedReader entrada = new BufferedReader(fr);
            String s;
            int flag = 0;
            String auxiliar = "";
            String ip="";
            String mac = "";
            while((s = entrada.readLine()) != null)
            {
                  //System.out.println(s);
                  if (s.toUpperCase().indexOf("LEASE")  != -1
|| s.toUpperCase().indexOf("HARDWARE ETHERNET") !=-1)
                  {
                        if   (s.toUpperCase().indexOf("LEASE")
!= -1)
                        {
                              if                            (
!(s.toUpperCase().indexOf("#") != -1 ))
                              {
                                    auxiliar = s + "  \n";
                                    flag=1;
                              }
                        }

                        if(flag==1)
                        {
                              if
(s.toUpperCase().indexOf("HARDWARE ETHERNET") !=-1)
                              {
                                    flag=2;
                              }
                        }

                        if ( !(s.toUpperCase().indexOf("#") !=
-1 ))
                        {
                              if (flag==2)
                              {
                                    if (!auxiliar.equals(""))
                                    {
```

```
                                                        texto += auxiliar;
                                                        ip                    =
auxiliar.substring(6, 21);
                                                        System.out.println("ip
- " +ip);
                                                }
                                                mac = s.substring(20,37);
                                                DatosIp    dat    =    new
DatosIp();
                                                vec.add(dat);
                                                dat.setIp(ip);
                                                dat.setMac(mac);
                                                System.out.println("mac  -  "
+ mac);
                                                texto += s + "  \n";
                                                flag=0;
                                        }
                                }
                        }
                }
                System.out.println(texto);
                 return(vec);
        }

        public static void main (String [] args)
        {

                try {
                        leerDhcp();
                } catch (Exception e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
        }

}
```

```
package com.general.util;

public class ValidacionesVarias {

                /** Valida la cedula o el ruc */
                public static boolean validarDocumento(String
numero)throws Exception
                {
                        boolean valor = true;
                        try {
                                int suma = 0;
                                int residuo = 0;
```

```
                        boolean privada = false;
                        boolean publica = false;
                        boolean natural = false;
                        int numeroProvincias = 22;
                        int digitoVerificador = 0;
                        int modulo = 11;

                        int d1, d2, d3, d4, d5, d6, d7, d8, d9,
d10;
                        int p1, p2, p3, p4, p5, p6, p7, p8, p9;

                        d1 = d2 = d3 = d4 = d5 = d6 = d7 = d8 = d9
= d10 = 0;
                        p1 = p2 = p3 = p4 = p5 = p6 = p7 = p8 = p9
= 0;

                        if (numero.length() < 10) {
                              valor = false;
                              throw new Exception("La
identificaciòn no tiene el nùmero de caractares validos");
                        }

                        // Los primeros dos digitos corresponden
al codigo de la provincia
                        int provincia =
Integer.parseInt(numero.substring(0, 2));

                        if (provincia <= 0 || provincia >
numeroProvincias) {
                              valor = false;
                              throw new Exception("El codigo de la
provincia es invalido");
                        }

                        // Aqui almacenamos los digitos de la
cedula en variables.
                        d1 = Integer.parseInt(numero.substring(0,
1));
                        d2 = Integer.parseInt(numero.substring(1,
2));
                        d3 = Integer.parseInt(numero.substring(2,
3));
                        d4 = Integer.parseInt(numero.substring(3,
4));
                        d5 = Integer.parseInt(numero.substring(4,
5));
                        d6 = Integer.parseInt(numero.substring(5,
6));
                        d7 = Integer.parseInt(numero.substring(6,
7));
                        d8 = Integer.parseInt(numero.substring(7,
8));
                        d9 = Integer.parseInt(numero.substring(8,
9));
```

```
                        d10 = Integer.parseInt(numero.substring(9,
10));

                        // El tercer digito es:
                        // 9 para sociedades privadas y
extranjeros
                        // 6 para sociedades publicas
                        // menor que 6 (0,1,2,3,4,5) para personas
naturales
                        if (d3 == 7 || d3 == 8) {
                              valor = false;
                              throw new Exception("El tercer
digito ingresado es invalido");
                        }


                        // Solo para personas naturales (modulo
10)
                        if (d3 < 6) {
                              natural = true;
                              modulo = 10;
                              p1 = d1 * 2;
                              if (p1 >= 10)
                                    p1 -= 9;
                              p2 = d2 * 1;
                              if (p2 >= 10)
                                    p2 -= 9;
                              p3 = d3 * 2;
                              if (p3 >= 10)
                                    p3 -= 9;
                              p4 = d4 * 1;
                              if (p4 >= 10)
                                    p4 -= 9;
                              p5 = d5 * 2;
                              if (p5 >= 10)
                                    p5 -= 9;
                              p6 = d6 * 1;
                              if (p6 >= 10)
                                    p6 -= 9;
                              p7 = d7 * 2;
                              if (p7 >= 10)
                                    p7 -= 9;
                              p8 = d8 * 1;
                              if (p8 >= 10)
                                    p8 -= 9;
                              p9 = d9 * 2;
                              if (p9 >= 10)
                                    p9 -= 9;
                        }

                        // Solo para sociedades publicas (modulo
11)
                        // Aqui el digito verficador esta en la
posicion 9, en las otras 2
```

```
                        // en la pos. 10
                        if (d3 == 6) {
                              publica = true;
                              p1 = d1 * 3;
                              p2 = d2 * 2;
                              p3 = d3 * 7;
                              p4 = d4 * 6;
                              p5 = d5 * 5;
                              p6 = d6 * 4;
                              p7 = d7 * 3;
                              p8 = d8 * 2;
                              p9 = 0;
                        }

                        /* Solo para entidades privadas (modulo
11) */
                        if (d3 == 9) {
                              privada = true;
                              p1 = d1 * 4;
                              p2 = d2 * 3;
                              p3 = d3 * 2;
                              p4 = d4 * 7;
                              p5 = d5 * 6;
                              p6 = d6 * 5;
                              p7 = d7 * 4;
                              p8 = d8 * 3;
                              p9 = d9 * 2;
                        }

                        suma = p1 + p2 + p3 + p4 + p5 + p6 + p7 +
p8 + p9;
                        residuo = suma % modulo;

                        // Si residuo=0, dig.ver.=0, caso
contrario 10 - residuo
                        digitoVerificador = residuo == 0 ? 0 :
modulo - residuo;
                        int longitud = numero.length(); //
Longitud del string

                        // ahora comparamos el elemento de la
posicion 10 con el dig. ver.
                        if (publica == true) {
                              if (digitoVerificador != d9) {
                                    valor = false;
                                    throw new Exception("El ruc de
la empresa del sector publico es incorrecto");
                              }
                              /* El ruc de las empresas del sector
publico terminan con 0001 */
                              if (!numero.substring(9,
longitud).equals("0001")) {
                                    valor = false;
                                    throw new Exception("El ruc de
```

```
la empresa del sector publico debe terminar con 0001");
                              }
                        }

                        if (privada == true) {
                              if (digitoVerificador != d10) {
                                    valor = false;
                                    throw new Exception("El ruc de
la empresa del sector privado es incorrecto");
                              }
                              if (!numero.substring(10,
longitud).equals("001")) {
                                    valor = false;
                                    throw new Exception("El ruc de
la empresa del sector privado debe terminar con 001");
                              }
                        }

                        if (natural == true) {
                              if (digitoVerificador != d10) {
                                    valor = false;
                                    throw new Exception("El numero
de cedula de la persona natural es incorrecto");
                              }
                              if (numero.length() > 10
                                          && !numero.substring(10,
longitud).equals(
                                                      "001")) {
                                    valor = false;
                                    throw new Exception("El ruc de
la persona natural debe terminar con 001");
                              }
                        }
                  } catch (Exception e) {
                        valor = false;
                        throw new Exception("Error " +
e.getMessage());
                  }
                  return valor;
            }
      }
```

```
WebRoot
    codebase
    css
    imagenes
    img
    js
    logosEmpresa
    META-INF
    paginas
    reportes
    Scripts
    swf
    WEB-INF
    flash.html
    index.html
    index1.jsp
    loginPrincipal.jsp
    MyHtml.html
    reportunoe.jsp
hibernate.reveng.xml
```

```
<%@ page language="java" pageEncoding="ISO-8859-1"%>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic"%>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>

 <%

 // Pantalla pricipal permite el acceso a la aplicacion

  //Variable de error
  //en caso de existir
  Exception error1 =
(Exception)request.getSession().getAttribute("ERRORGENERAL");
  %>
<html>
<head>
<title>BEMUSOFT - V1.0</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
<style type="text/css">
<!-- permite la creacion de los estilos en la pantalla del login -->
<!--
body,td,th {
        font-family: Verdana, Arial, Helvetica, sans-serif;
}
body {
        margin-left: 0px;
        margin-top: 0px;
```

```
        margin-right: 0px;
        margin-bottom: 0px;
}
a:link {
        color: #FFFFFF;
        text-decoration: none;
}
a:visited {
        text-decoration: none;
        color: #FFFFFF;
}
a:hover {
        text-decoration: none;
        color: #80B7DE;
}
a:active {
        text-decoration: none;
        color: #00FF00;
}
.Estilo3 {
        color: white;
        font-weight: bold;
        font-size: 12px;
}
.Estilo5 {
        font-size: 12px;
        font-weight: bold;
        color: NAVY;
}
.tablaLoggin {
        background-attachment: scroll;
        background-image: url(.../imagenes/marco.gif);
        background-repeat: no-repeat;
        background-position: center center;
}
.Estilo7 {color: #175CB7}
-->
</style>
<%
// Import de las clases y archivos para la pagina del login -->
// se importan los colores, las validaciones y los efectos -->
%>
<script type="text/javascript"
src="<%=request.getContextPath()%>/js/general.js"></script>
<link rel="stylesheet"
href="<%=request.getContextPath()%>/css/disenioGeneral.css"
type="text/css"></link>
<script type="text/javascript" language="Javascript1.1"
src="<%=request.getContextPath()%>/js/validacionesStruts.js"></scrip
t>
<html:javascript formName="loginForm" staticJavascript="false"/>
</head>
<script type="text/javascript">
function habilitarEnter(e)
```

```
{
        //fuincion para habilitar el enter
        var tecla=event.keyCode;
        if(parseInt(tecla)==13)
        {
           validarCampos();
        }
}

function validarCampos()
{
    ;
}
</script>
           <%
           if (error1 != null)
            if (error1.getMessage() != null)
            {
           {%>
           <script type="text/javascript">
              //permite presentar el error
              //con el dialogo

        window.showModalDialog("<%=request.getContextPath()%>/
error.do", "_blank", "dialogheight=20" )
           </script>
           <%}
           }
           %>
        <!-- creacion de la forma y el cuerpo de la pagina -->
        <body onkeypress="habilitarEnter(event);"
bgcolor="7EAF39">
            <!-- Con esto se hace referencia al action del
controlador que se va a usar -->
           <html:form action="/login" focus="usuario"
onsubmit="return validateLoginForm(this)">
           <table width="100%" border="0" cellpadding="0"
cellspacing="0" bgcolor="7EAF39" height="30%">
                <tr>
                    <td width="111" height="60"> </td>

                      <td width="121"> </td>
                </tr>
                <tr>
                <td height="125"> </td>
                <td valign="top" width="15%">
                           <fieldset >
                           <table class="tablaLoggin"
width="70%" border="0" cellpadding="0" cellspacing="0"  >
                           <!--DWLayoutTable-->
```

```
                            <tr>
                                 <td width="16"></td>
                                      <td height="18"
colspan="3">
                                           <FONT
face="Verdana, Arial, Helvetica, sans-serif" size=2>

        <logic:messagesPresent>

        <html:messages id="error">

        <%=error%>

        </html:messages>

        </logic:messagesPresent>
                                           </FONT>
                                      </td>
                                      <td width="18"></td>
                            </tr>
                            <tr>
                                 <td></td>
                                 <td width="85" height="22"
valign="middle"><span class="Estilo3"><bean:message
key="etiquetas.login.user"/>:</span></td>
                                 <td colspan="2" valign="top">
                                      <html:text
property="usuario"
onfocus="cambiarEstiloCajaTexto(event,'on')"
onblur="cambiarEstiloCajaTexto(event,'off')"
styleClass="cajaTexto" size="19"/>
                                      </td>
                                 <td></td>
                            </tr>
                            <tr>
                                 <td></td>
                                 <td height="22"
valign="middle"><span class="Estilo3"><bean:message
key="etiquetas.login.clave"/>:</span></td>
                                 <td colspan="2" valign="top">
                                      <html:password
property="clave" onfocus="cambiarEstiloCajaTextoPass(event,'on')"
onblur="cambiarEstiloCajaTextoPass(event,'off')"
styleClass="cajaTextoPass" size="19"/>
                                 <td></td>
                            </tr>
                            <tr valign="middle">
                                 <td></td>
                                 <td colspan="2" align="center"
valign="middle">
                                      <br>
```

```
                                                <input type="Reset"
value="Cancelar" class="boton">  <html:submit
value="Ingresar" styleClass="boton"/>
                                          </td>
                                          <td></td>
                              </tr>
                              <tr>
                                    <td></td>
                                    <td height="18" colspan="3"><!--
DWLayoutEmptyCell--> </td>
                                    <td></td>
                              </tr>
                  <tr>
                              <td></td>
                              <td height="1"></td>
                              <td width="99"></td>
                              <td></td>
                              <td></td>
                  </tr>
            </table>
            </fieldset>
        </td>
            <td> </td>
            </tr>
            <tr>
    <td height="18"> </td>
    <td> </td>
    <td> </td>
  </tr>

</table>
<table width="100%" height="60%" border="0" cellpadding="0"
cellspacing="0" background="images/rayamitad.gif" bgcolor="white">
  <!--DWLayoutTable-->
  <tr>
    <td width="100%" height="254" valign="middle"><div
align="center">
    <img src="imagenes/bemusSoft.JPG"></img>
    </div></td>
  </tr>
</table>
</html:form>
        </body>
</html>
```

```
<%@ page language="java" pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean"
prefix="bean"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html"
prefix="html"%>
<%@ page import="com.general.modelo.persistentes.*" %>
 <%
  Exception error =
```

```
(Exception)request.getSession().getAttribute("ERRORGENERAL");
  Personas persona =
(Personas)request.getSession().getAttribute("PERSONAENCONTRADA"
);
  String   datoIden =
(String)request.getSession().getAttribute("IDENTIFICACION");
  String   usuario =
(String)request.getSession().getAttribute("DESCUSUARIO");
  request.getSession().setAttribute("PERSONAENCONTRADA", null);
  request.getSession().setAttribute("IDENTIFICACION", null);
  request.getSession().setAttribute("DESCUSUARIO", null);
  datoIden = datoIden==null?"":datoIden;
  usuario = usuario==null?"":usuario;
  String nombreCompleto = "";
  String idPersona = "";
  if (persona != null)
  {
        nombreCompleto =
persona.getPrimerNombre().toUpperCase() + " " +
persona.getSegundoNombre().toUpperCase() + " " +
persona.getApellidoPaterno().toUpperCase() + " " +
persona.getApellidoMaterno().toUpperCase();
        idPersona = persona.getIdPersona().toString();
        if (datoIden.equals(""))
        {
           datoIden = persona.getCedula();
           datoIden = datoIden==null?"":datoIden;
        }
        if (datoIden.equals(""))
        {
           datoIden = persona.getCedula();
           datoIden = datoIden==null?"":datoIden;
        }

        if (datoIden.equals(""))
        {
           datoIden = persona.getPasaporte();
           datoIden = datoIden==null?"":datoIden;
        }
  }

  %>

<html>
        <head>
           <title>JSP for IngresoUsuarioForm form</title>
        <link rel="stylesheet"
href="<%=request.getContextPath()%>/css/estiloFormas.css"
type="text/css"></link><script type="text/javascript"
```

```
src="<%=request.getContextPath()%>/js/general.js"></script>
        <script type="text/javascript"
language="Javascript1.1"
src="<%=request.getContextPath()%>/js/validacionesStruts.js"></
script>
        <script type="text/javascript">
         function limpiar()
     {

        window.open("<%=request.getContextPath()%>//paginas/tr
ansacciones/seguridad/ingresoUsuario.jsp","TRABAJO");

     }

           function buscarCedula(valor)
           {
                var dato = valor;
           var ruta ="";
                if (dato!="")
           {

                     ruta =
"<%=request.getContextPath()%>/buscarIdentificacionPersona.do?d
atoIdentificacion="+dato+"&accion=1";
                window.open(ruta, "_self");
               }
                else
               {
                alert("Ingrese el numero de Identificacion");
                }
           }


           function validarSiNumero(numero, obj)
        {
           if (!/^([0-9])*$/.test(numero))
           {
                alert("Las identificaciones deben ser
numericas");
                obj.value = "";
                obj.focus;
           }
        }

        </script>

        </head>
        <body>
        <%
```

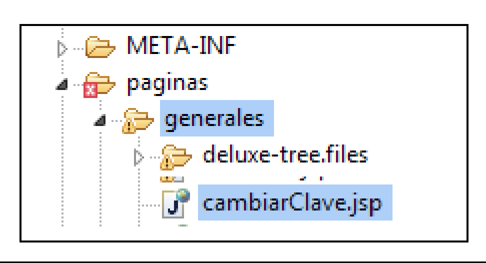
```
            if (error != null)
            {%>
            <script type="text/javascript">

        window.showModalDialog("<%=request.getContextPath()%>/
error.do", "_blank", "dialogheight=20" )
            </script>
            <%}%>
            <html:form action="/ingresoUsuario">
            <table align="center" width="85%" cellspacing="1"
cellpadding="1"  class="tabla" >
                 <tr>
                      <td colspan="4"
class="cabeceraForma">Creacion De Usuarios</td>
                 </tr>
                 <tr height="5">
                      <td colspan="4"></td>
                 </tr>
                 <tr>
                      <td colspan="4"
class="titulo"> Datos Usuario</td>
                 </tr>

                 <tr>
                      <td class="etiquetaForma"
width="20%">Cedula:*</td>
                      <td class="detalleForma">
                           <html:text property="cedula"
value="<%=datoIden%>"
onfocus="cambiarEstiloCajaTexto(event,'on')"
onblur="cambiarEstiloCajaTexto(event,'off')"
onchange="validarSiNumero(this.value, this);"
styleClass="cajaTexto"/><html:errors
property="descripcionRol"/> <img
src="<%=request.getContextPath()%>/imagenes/kedit.png"
onclick="buscarCedula(cedula.value);" alt="Buscar" width="20px"
height="20px" align="left" border="1" hspace="0"
vspace="0"></img>
                      </td>
                      <td class="etiquetaForma"
width="20%">Nombre Completo:*</td>
                      <td class="detalleForma" width="30%">
                           <html:textarea
property="nombreCompleto" value="<%=nombreCompleto%>"
onfocus="cambiarEstiloCajaTexto(event,'on')"
onblur="cambiarEstiloCajaTexto(event,'off')"
styleClass="cajaTexto" readonly="true" rows="2"
cols="31"/><html:errors property="descripcionRol"/>
                      </td>
```

```
                 </tr>
                 <tr>
                 <td class="etiquetaForma"
width="20%">Usuario:*</td>
                      <td class="detalleForma" width="30%">
                  <html:hidden property="idPersona"
value="<%=idPersona%>"/><html:errors property="idPersona"/>
                  <html:text property="descripcionUsuario"
value="<%=usuario.toUpperCase() %>"
onfocus="cambiarEstiloCajaTexto(event,'on')"
onblur="cambiarEstiloCajaTexto(event,'off')"
styleClass="cajaTexto" readonly="true" /><html:errors
property="descripcionUsuario"/>
                      </td>
                      <td class="etiquetaForma"
width="20%">Rol:*</td>
                      <td class="detalleForma" width="30%">
                  <html:hidden
property="idPersona"/><html:errors property="idPersona"/>
                  <html:select property="rol"
onfocus="cambiarEstiloCombo(event,'on')"
onblur="cambiarEstiloCombo(event,'off')" styleClass="combo">
                  <html:option value="-
1">SELECCIONE</html:option>
                      <html:options collection="ROLESSISTEMAS"
property="idRol" labelProperty="descRol"/>
                  </html:select>
                  <html:errors property="rol"/>
                      </td>
                 </tr>
                 <tr>
                      <td class="etiquetaForma"
width="20%">Observacion:*</td>
                      <td class="detalleForma" width="30%"
colspan="3">
                      <html:textarea property="observacion"
rows="2" cols="130"
onfocus="cambiarEstiloCajaTexto(event,'on')"
onblur="cambiarEstiloCajaTexto(event,'off')"
styleClass="cajaTexto"/><html:errors property="observacion"/>
                      </td>
                 </tr>
                 <tr height="40">
                      <td class="pieForma" colspan="4"
align="center" height="40">
                           <input type="Reset" value="Limpiar"
class="boton" onclick="limpiar();" />
                           <html:submit value="Aceptar"
styleClass="boton" />
```

```
                        </td>
                  </tr>
            </table>
            </html:form>
        </body>
</html>
```




```
<%@ page language="java"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic"%>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@page import="com.general.modelo.persistentes.Usuarios"%>
<%@page import="com.general.util.Sesion"%>
<%@page import="com.general.util.*"%>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>BEMUS-SOFT SEGURIDAD - Cambio de Clave</title>
<%
          Exception error =
(Exception)request.getSession().getAttribute("ERRORGENERAL");
        Usuarios
l_usuario=(Usuarios)session.getAttribute(Sesion.USUARIO);
        String   lsInicio = request.getParameter("inicial");
        lsInicio = lsInicio==null?"":lsInicio;
        String   lsClaveActual = l_usuario.getClave();
%>
<link rel="stylesheet"
href="<%=request.getContextPath()%>/css/estiloFormas.css"
type="text/css"></link>
<script type="text/javascript"
src="<%=request.getContextPath()%>/js/general.js"></script>
<script type="text/javascript" language="Javascript1.1"
src="<%=request.getContextPath()%>/js/validacionesStruts.js"></script
>
<html:javascript formName="formaCambioClave"
staticJavascript="false"/>
<script type="text/javascript">
```

```
function validarClaveActual(object)
{
        var claveIngresada = object.value;
        var claveActual =
"<%=UtilCryptography.desencriptar(lsClaveActual)%>";
        if (claveIngresada != claveActual)
        {
            alert("La clave que ingreso no es igual a su clave
actual");
            object.value="";
            return;
        }
}

function validarClaveNueva(object)
{
        var claveConfNuevaIngresada = object.value;
        var claveNuevaIngresada = formaCambioClave.claveNueva.value;

        if (claveNuevaIngresada=="")
        {
            alert("Ingrese primero la clave nueva");
            formaCambioClave.claveNueva.value = "";
            formaCambioClave.confirmacionClaveNueva.value = "";
            return;
        }

        if (claveConfNuevaIngresada != claveNuevaIngresada)
        {
            alert("Las claves nuevas no concuerdan, por favor
ingresarlas correctamente");
            object.value="";
            formaCambioClave.claveNueva.value = "";
            return;
        }
}

function verificarInicial()
{
        var inicial = "<%=lsInicio%>";
        if (inicial == "S")
        {

        window.open("<%=request.getContextPath()%>/trabajo.do","TRABA
JO");
        }
}
</script>
</head>
<body topmargin="0" bottommargin="0" leftmargin="0" rightmargin="0"
onunload="verificarInicial();">

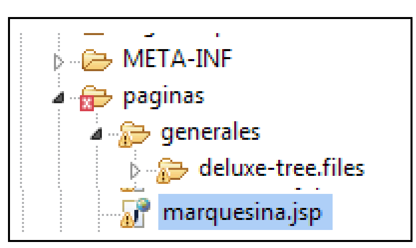
<%
            if (error != null)
```

```
            {%>
            <script type="text/javascript">

        window.showModalDialog("<%=request.getContextPath()%>/error.d
o", "_blank", "dialogheight=20" )
            </script>
            <%}%>

<html:form action="/cambiarClave" onsubmit="return
validateFormaCambioClave(this)">
        <table border="0" width="100%" height="80%">
            <tr align="center" valign="middle">
            <td>
            <table align="center" width="85%" cellspacing="1"
cellpadding="1"  class="tabla" >
                  <tr>
                        <td colspan="4"
class="cabeceraForma">Administracion De Claves</td>
                  </tr>
                  <tr height="5">
                        <td colspan="4"></td>
                  </tr>
                  <tr>
                        <td colspan="2" class="titulo">CAMBIAR
CLAVE</td>
                  </tr>
                  <tr>
                        <td class="etiquetaForma"
width="32%"><bean:message key="etiquetas.cambiarClave.user"/> *:
</td>
                        <td class="detalleForma"><html:text
property="user" maxlength="20" size="30" readonly="true"
onfocus="cambiarEstiloCajaTexto(event,'on')"
onblur="cambiarEstiloCajaTexto(event,'off')" styleClass="cajaTexto"
value="<%=l_usuario.getDescUsuario()%>"></html:text></td>
                  </tr>
                  <tr>
                        <td class="etiquetaForma"
width="35%"><bean:message key="etiquetas.cambiarClave.claveActual"/>
*:</td>
                        <td class="detalleForma"><html:password
property="claveActual" maxlength="20" size="30"
onfocus="cambiarEstiloCajaTextoPass(event,'on')"
onblur="cambiarEstiloCajaTextoPass(event,'off')"
styleClass="cajaTextoPass"
onchange="validarClaveActual(this);"></html:password></td>
                  </tr>
                  <tr>
                        <td class="etiquetaForma"
width="35%"><bean:message key="etiquetas.cambiarClave.claveNueva"/>
*:</td>
                        <td class="detalleForma"><html:password
property="claveNueva" maxlength="20" size="30"
onfocus="cambiarEstiloCajaTextoPass(event,'on')"
```

```
onblur="cambiarEstiloCajaTextoPass(event,'off')"
styleClass="cajaTextoPass" ></html:password></td>
                  </tr>
                  <tr>
                        <td class="etiquetaForma"
width="35%"><bean:message
key="etiquetas.cambiarClave.confirmacionClaveNueva"/> *:</td>
                        <td class="detalleForma"><html:password
property="confirmacionClaveNueva" maxlength="20" size="30"
onfocus="cambiarEstiloCajaTextoPass(event,'on')"
onblur="cambiarEstiloCajaTextoPass(event,'off')"
styleClass="cajaTextoPass" onchange="validarClaveNueva(this);"
></html:password></td>
                  </tr>
                  <tr class="pieForma">
                        <td colspan="2" align="center" >
                              <html:submit value="Guardar"
styleClass="boton"/>
                        </td>
                  </tr>
            </table>
            </td>
            </tr>
        </table>
</html:form>
</body>
</html>
```




```
<%@ page import="com.general.modelo.persistentes.*" %>
<%@page import="java.util.*" %>
  <%
  Empresas empresa  =
(Empresas)request.getSession().getAttribute("EMPRESA");
  List mensajesMarquesina =
(List)request.getSession().getAttribute("PARAMETROSMAR");
  if (empresa==null)
  {
        empresa = new Empresas();
        empresa.setRazonSocial("BEMUSOFT");
  }

  %>
```

```
<html>
<head>
<title>Marquesina con efecto disolvente</title>
<style>
.etiquetasLink
{
	text-decoration: underline;
	font-size: 9pt;
	font-weight: bold;
	color:#00224C;
	letter-spacing: 1px;
	font-family: Arial;
	color:#00224C;
	font-weight:bold;
	vertical-align: top;
}

.backFrame
{
	background-color: #999999;
}

</style>

<!--<p align="center"><b><font
class="etiquetasLink">Anuncios</font></b><br></p>-->
<script LANGUAGE="JavaScript">

// Edit the variables below:

// Your messages. Add as many as you like
var message=new Array()
message[0]="Sistema De Facturacion Y Administracion De ISP, Gestion
En La Empresa <%=empresa.getRazonSocial().toUpperCase()%>";
message[1]="Grupo De Seminario Tema REDES";




// the URLs of your messages
var messageurl=new Array()
/*messageurl[0]="http://service.bfast.com/bfast/click?bfmid=7279965&
siteid=30125004&bfpage=webclipart3"
messageurl[1]="http://service.bfast.com/bfast/click?bfmid=7279965&si
teid=30125004&bfpage=javascript"
messageurl[2]="http://service.bfast.com/bfast/click?bfmid=599904&sit
eid=30259589&bfpage=gifbot"*/

// the targets of the links
// accepted values are '_blank' or '_top' or '_parent' or '_self'
// or the name of your target-window (for instance 'main')
var messagetarget=new Array()
/*messagetarget[0]="_blank"
messagetarget[1]="_blank"
```

```
messagetarget[2]="_blank"*/

// font-color
var messagecolor= new Array()

messagecolor[0]="navy"
messagecolor[1]="navy"


// distance of the scroller to the left margin of the browser-window
(pixels)
var scrollerleft=10

// distance of the scroller to the top margin of the browser-window
(pixels)
var scrollertop=10

// width of the scroller (pixels)
var scrollerwidth=1200

// height of the scroller (pixels)
var scrollerheight=40

// speed 1: lower means faster
var pause=25

// speed 2: higher means faster
var step=2

// font-size
var fntsize=14


// font-family
var fntfamily="Arial"

// font-weight: 1 means bold, 0 means normal
var fntweight=1

// do not edit the variables below
var fadeimgwidth=40
var fadeimgleftcontent,fadeimgrightcontent
var clipleft,clipright,cliptop,clipbottom
var i_message=0
var timer
var textwidth
var textcontent=""
if (fntweight==1) {fntweight="700"}
else {fntweight="100"}

function init() {
        gettextcontent()

        fadeimgleftcontent="<img src='fadeimgleft.gif'
```

```
width="+fadeimgwidth+" height="+scrollerheight+">"
        fadeimgrightcontent="<img src='fadeimgright.gif'
width="+fadeimgwidth+" height="+scrollerheight+">"

    if (document.all) {
            text.innerHTML=textcontent
            fadeimgleft.innerHTML=fadeimgleftcontent
            fadeimgright.innerHTML=fadeimgrightcontent
            textwidth=text.offsetWidth
            document.all.text.style.posTop=scrollertop
        document.all.text.style.posLeft=scrollerleft+scrollerwidth
            document.all.fadeimgleft.style.posTop=scrollertop
        document.all.fadeimgleft.style.posLeft=scrollerleft
            document.all.fadeimgright.style.posTop=scrollertop

document.all.fadeimgright.style.posLeft=scrollerleft+scrollerwidth-
fadeimgwidth

            clipleft=0
            clipright=0
            cliptop=0
            clipbottom=scrollerheight
            document.all.text.style.clip ="rect("+cliptop+"
"+clipright+" "+clipbottom+" "+clipleft+")"
        scrolltext()
    }
        if (document.layers) {
            document.text.document.write(textcontent)
            document.text.document.close()
            document.fadeimgleft.document.write(fadeimgleftcontent)
            document.fadeimgleft.document.close()

        document.fadeimgright.document.write(fadeimgrightcontent)
            document.fadeimgright.document.close()
            textwidth=document.text.document.width
            document.text.top=scrollertop
            document.text.left=scrollerleft+scrollerwidth
            document.fadeimgleft.top=scrollertop
        document.fadeimgleft.left=scrollerleft
            document.fadeimgright.top=scrollertop
        document.fadeimgright.left=scrollerleft+scrollerwidth-
fadeimgwidth

            document.text.clip.left=0
            document.text.clip.right=0
            document.text.clip.top=0
            document.text.clip.bottom=scrollerheight

        scrolltext()
    }
}

function scrolltext() {
    if (document.all) {
```

```
            if (document.all.text.style.posLeft>=scrollerleft-
textwidth) {
                  document.all.text.style.posLeft-=step
                  clipright+=step
                  if (clipright>scrollerwidth) {
                        clipleft+=step
                  }
                  document.all.text.style.clip ="rect("+cliptop+"
"+clipright+" "+clipbottom+" "+clipleft+")"

                  var timer=setTimeout("scrolltext()",pause)
            }
            else {
                  changetext()
            }
        }
   if (document.layers) {
            if (document.text.left>=scrollerleft-textwidth) {
                  document.text.left-=step
                  document.text.clip.right+=step
                  if (document.text.clip.right>scrollerwidth) {
                        document.text.clip.left+=step
                  }
                  var timer=setTimeout("scrolltext()",pause)
            }
            else {
                  changetext()
            }
        }
}

function changetext() {
    i_message++
        if (i_message>message.length-1) {i_message=0}
        gettextcontent()
        if (document.all) {
            text.innerHTML=textcontent
            textwidth=text.offsetWidth

        document.all.text.style.posLeft=scrollerleft+scrollerwidth
            clipleft=0
            clipright=0
            document.all.text.style.clip ="rect("+cliptop+"
"+clipright+" "+clipbottom+" "+clipleft+")"

        scrolltext()
        }

        if (document.layers) {
            document.text.document.write(textcontent)
            document.text.document.close()
            textwidth=document.text.document.width

            document.text.left=scrollerleft+scrollerwidth
```

```
            document.text.clip.left=0
            document.text.clip.right=0

        scrolltext()
        }
}

function gettextcontent() {
        textcontent="<span style='position:relative;font-
size:"+fntsize+"pt;font-family:"+fntfamily+";font-
weight:"+fntweight+"'>"
        //textcontent+="<a href="+messageurl[i_message]+"
target="+messagetarget[i_message]+">"
        textcontent+="<nobr><font
color="+messagecolor[i_message]+">"+message[i_message]+"</font></nob
r></a></span>"
}

window.onresize=init;

// - End of JavaScript - -->
</script>
<style>
A {
text-decoration:none;
}
A:Hover {
text-decoration:underline;
color:#00224C;
}
body{
overflow-x:hidden;
overflow-y:scroll;
}


</style>
</head>

<body  onLoad="init()">

<!--codigo java script para la marquesina se encuentra detallado
abajo-->


<div ID="text" style="position:absolute;left:-2000px; background-
color:#FFFFFF" >
</div>
<div ID="fadeimgleft">
  </div>
<div ID="fadeimgright" style="position:absolute;left:-1000px">
  </div>

 <div id="imagne"  align="left">
```

```
 <br>
 </div>

 <!--
 <center>
 <font  color="navy" size="6">hola mundo</font>
 </center>
 <hr width="100%" size="2">-->
</body>

</html>
```

Herramientas para creacion de Scripts en Linux

Perl.-

Está basado en un estilo de bloques como los del C o AWK, y fue ampliamente adoptado por su destreza en el procesado de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

Tiene caracteristicas que soportan una variedad de paradigmas de programación, como la imperativa, funcional y la orientada a objeros. Al mismo tiempo, no obliga a seguir ningún paradigma en particular, ni obliga al programador a elegir alguna de ellas.

Modulo DBI

El módulo DBI proporciona una forma unificada, una API, de acceder a las bases de datos; es decir, que independientemente del gestor de bases de datos que usemos, usaremos las mismas funciones para hacer inserciones, actualizaciones, consultas, etc. Están soportadas tanto bases de datos "Open Source" como Postgres, miniSQL o MySQL, como algunas comerciales como Oracle, Sybase o Informix.

Script Planes

```
#!/usr/bin/perl

use DBI;

@datos = ("DEVICE=eth0,10Mbit,1Mbit", "RATE=", "WEIGHT=", "PRIO=5");
```

```
$db = 'DBI:mysql:facturacionisp:127.0.0.1';
$username = 'root';
$pass = 'redes';
$k="Kbit";

#$sth3->execute;

#################################################################
$fechadia = localtime time;
@valuesfecha = split(' ',$fechadia);
#print $values[1],"-", $values[2],"-", $values[4];
$mesmaq = $valuesfecha[1];

if ($mesmaq eq 'Jan'){ $mes=1; }
if ($mesmaq eq 'Feb'){ $mes=2; }
if ($mesmaq eq 'Mar'){ $mes=3; }
if ($mesmaq eq 'Apr'){ $mes=4; }
if ($mesmaq eq 'May'){ $mes=5; }
if ($mesmaq eq 'Jun'){ $mes=6; }
if ($mesmaq eq 'Jul'){ $mes=7; }
if ($mesmaq eq 'Agu'){ $mes=8; }
if ($mesmaq eq 'Sep'){ $mes=9; }
if ($mesmaq eq 'Oct'){ $mes=10;}
if ($mesmaq eq 'Nov'){ $mes=11;}
if ($mesmaq eq "Dec"){ $mes=12;}

$fecha2= $valuesfecha[4].'-'.$mes.'-'.$valuesfecha[2];
#print $fecha1;
#################################################################
$dbh = DBI->connect($db, $username, $pass);
$sql = qq{ SELECT * FROM productos WHERE estado = 'C'};
$dbh->do($sql);
$sth = $dbh->prepare($sql);
$sth->execute;

while(@array = $sth->fetchrow_array){
$dbh2 = DBI->connect($db, $username, $pass);
$sql2 = qq{ SELECT * FROM numeros ORDER BY id DESC };
$dbh2->do($sql2);
$sth2 = $dbh2->prepare($sql2);
$sth2->execute;
@array1 = $sth2->fetchrow_array;
$valor = $array1[0];
$valor1 = $valor + 1;
$dbh3 = DBI->connect($db, $username, $pass);
$sql3 = qq{ INSERT INTO numeros (`id`) VALUES ('$valor1'); };
$dbh3->do($sql3);
```

```
$idpro = $array[0];
$nombre = $array[2];
$valormax = $array[7];
$nombrefile = "cbq-00" . $valor . "." . $nombre;
$ruta = "+>/etc/sysconfig/cbq/".$nombrefile;
#print $ruta;

$wei=$valormax/10;
$weifinal1 = int($wei);

open(PLANES,$ruta) || die "No pudo abrirse: $!";
print PLANES $datos[0], "\n";
print PLANES $datos[1] . $valormax . $k, "\n";
print PLANES $datos[2] . $weifinal1 . $k, "\n";
print PLANES $datos[3], "\n";
close(PLANES);

$dbh = DBI->connect($db, $username, $pass);
$sql4 = qq{ UPDATE provisiona_activacion_plan SET estado = 'A',
fecha_actualiza = '$fecha2' WHERE codi_plan ='$nombre' };
$dbh->do($sql4);
$sth4 = $dbh->prepare($sql4);
$sth4->execute;

$dbh = DBI->connect($db, $username, $pass);
$sql5 = qq{ UPDATE productos SET estado = 'A', fecha_actualizacion = '$fecha2',
nombrefile = '$nombrefile' WHERE id_producto ='$idpro' AND estado = 'C'};
$dbh->do($sql5);
$sth5 = $dbh->prepare($sql5);
$sth5->execute;

}
```

Script Reserva

```
#!/usr/bin/perl

use DBI;

@datos = ("RULE=");

$db = 'DBI:mysql:facturacionisp:127.0.0.1';
$username = 'root';
$pass = 'redes';

$ruta1= ">/etc/sysconfig/cbq/test";
open(TEST,$ruta1) || die "No pudo abrirse: $!";

$dbh = DBI->connect($db, $username, $pass);
```

```
$sql = qq{ SELECT * FROM servicios_contratados WHERE estado = 'R' AND
valor_referencia_1 != 'NULL' AND valor_referencia_2 != 'NULL' AND
valor_referencia_1 !=
 'SERVCORREO' };
$dbh->do($sql);
$sth = $dbh->prepare($sql);
$sth->execute;

while(@array = $sth->fetchrow_array){
$ip = $array[8];
$contrato = $array[1];
$servicios = $array[0];

`iptables -t nat -A POSTROUTING -o eth0 -s $ip/32 -j MASQUERADE`;
$sq10 = qq{ INSERT INTO nat (ip) VALUES ('$ip') };
$dbh->do($sq10);
$sth10 = $dbh->prepare($sq10);
#$sth10->execute;

#print "UPDATE servicios_contratados SET estado = 'A' WHERE estado = 'R' AND
id_servicio = '$servicios' AND id_contrato = '$contrato'";
$sq2 = qq{ UPDATE servicios_contratados SET estado = 'A' WHERE estado = 'R'
AND id_servicio = '$servicios' AND id_contrato = '$contrato' };
$dbh->do($sq2);
$sth1 = $dbh->prepare($sq2);
$sth1->execute;

#print "SELECT * FROM contratos WHERE id_contrato = '$contrato'";
$sq3 = qq{ SELECT * FROM contratos WHERE id_contrato = '$contrato' };
$dbh->do($sq3);
$sth3 = $dbh->prepare($sq3);
$sth3->execute;
while(@array3 = $sth3->fetchrow_array){
$nombreplan = $array3[1];

#print "SELECT * FROM productos WHERE descripcion = '$nombreplan'";
$sq4 = qq{ SELECT * FROM productos WHERE descripcion = '$nombreplan' };
$dbh->do($sq4);
$sth4 = $dbh->prepare($sq4);
$sth4->execute;
while(@array4 = $sth4->fetchrow_array){
print TEST $datos[0] . $ip . "/32", "\n";
close(TEST);

$nombreplanfile = $array4[18];
$ruta="/etc/sysconfig/cbq/".$nombreplanfile;

print "more /etc/sysconfig/cbq/test >> $ruta";
```

```
`more /etc/sysconfig/cbq/test >> $ruta`;
`rm -rf /etc/sysconfig/cbq/test`;
`/sbin/cbq restart`;
}

}

}

$dbh = DBI->connect($db, $username, $pass);
$sq6 = qq{ SELECT * FROM servicios_contratados WHERE estado = 'R' AND
valor_referencia_3 = 'CORREO' AND valor_referencia_2 != 'NULL' };
$dbh->do($sq6);
$sth6 = $dbh->prepare($sq6);
$sth6->execute;

while(@array6 = $sth6->fetchrow_array){

$correo = $array6[8];
$id_servicio = $array6[0];
$id_contrato = $array6[1];

$cryptpw = crypt($correo, 14);

system("useradd -c $correo -s /bin/nologin -p $cryptpw $correo");

$sq7 = qq{ UPDATE servicios_contratados SET estado = 'A' WHERE estado = 'R'
AND id_servicio = '$id_servicio' AND valor_referencia_3 = 'CORREO' AND id_contrat
o = '$id_contrato' };
$dbh->do($sq7);
$sth7 = $dbh->prepare($sq7);
$sth7->execute;
}
```

Script Activar_lotes

```
#!/usr/bin/perl

use DBI;

$db = 'DBI:mysql:facturacionisp:127.0.0.1';
$username = 'root';
$pass = 'redes';


$dbh = DBI->connect($db, $username, $pass);
$sql = qq{ SELECT * FROM servicios_contratados WHERE estado = 'I' AND
descripcion_servicio != 'CORREO' };
$dbh->do($sql);
```

```
$sth = $dbh->prepare($sql);
$sth->execute;

while(@array = $sth->fetchrow_array){
$var=0;
$contrato = $array[1];
$ip = $array[8];
print "SELECT * FROM facturas WHERE id_contrato = '$contrato' AND estado = 'I'";
print "\n";
$sq2 = qq{ SELECT * FROM facturas WHERE id_contrato = '$contrato' AND estado = 'I'};
$dbh->do($sq2);
$sth1 = $dbh->prepare($sq2);
$sth1->execute;
#print $sth1;
print $var;
while(@array3 = $sth1->fetchrow_array){
$var=1;
print $var
print "entra al while";
}
if ($var eq '0')
{
print "entra if";
`iptables -t nat -A POSTROUTING -o eth0 -s $ip/32 -j MASQUERADE`;

$sq10 = qq{ INSERT INTO nat (ip) VALUES ('$ip') };
$dbh->do($sq10);
$sth10 = $dbh->prepare($sq10);
$sth10->execute;

$dbh = DBI->connect($db, $username, $pass);
$sq16 = qq{ UPDATE servicios_contratados SET estado = 'A', valor_referencia_3 = 'Automatico_reactivado'  WHERE estado = 'I' AND id_contrato = '$contrato'};
$dbh->do($sq16);
$sth16 = $dbh->prepare($sq16);
$sth16->execute;
}
}
```

Script Activar

```
#!/usr/bin/perl

use DBI;

$db = 'DBI:mysql:facturacionisp:127.0.0.1';
$username = 'root';
```

```
$pass = 'redes';

#$now = localtime time;
#@values = split(' ',$now);
#print $values[1],"-", $values[2],"-", $values[4];
#$mesmaq = $values[1];

$dbh = DBI->connect($db, $username, $pass);
$sql = qq{ SELECT * FROM reactivacion_servicios WHERE estado = 'I' };
$dbh->do($sql);
$sth = $dbh->prepare($sql);
$sth->execute;

#$fecha1= $values[4].'-'.$mes.'-'.'01';
#print $fecha1;

while(@array = $sth->fetchrow_array){
$idac = $array[0];
$ip = $array[3];
`iptables -t nat -A POSTROUTING -o eth0 -s $ip/32 -j MASQUERADE`;
$sq10 = qq{ INSERT INTO nat (ip) VALUES ('$ip') };
$dbh->do($sq10);
$sth10 = $dbh->prepare($sq10);
$sth10->execute;
$sq2 = qq{ UPDATE reactivacion_servicios SET estado = 'E' WHERE estado = 'I'
AND id_reactivacion = '$idac' AND ip_maquina = '$ip'};
$dbh->do($sq2);
$sth1 = $dbh->prepare($sq2);
$sth1->execute;

$sq16 = qq{ UPDATE servicios_contratados SET estado = 'A', valor_referencia_3 =
'Automatico_reactivado'  WHERE estado = 'I' AND valor_referencia_2 = '$ip' AN
D descripcion_servicio != 'CORREO'};
$dbh->do($sq16);
$sth16 = $dbh->prepare($sq16);
$sth16->execute;
}
```

Script Corte_lotes

```
#!/usr/bin/perl

use DBI;

$db = 'DBI:mysql:facturacionisp:127.0.0.1';
$username = 'root';
$pass = 'redes';

$now = localtime time;
```

```
@values = split(' ',$now);
#print $values[1],"-", $values[2],"-", $values[4];
$mesmaq = $values[1];

if ($mesmaq eq 'Jan'){ $mes=01; }
if ($mesmaq eq 'Feb'){ $mes=02; }
if ($mesmaq eq 'Mar'){ $mes=03; }
if ($mesmaq eq 'Apr'){ $mes=04; }
if ($mesmaq eq 'May'){ $mes=05; }
if ($mesmaq eq 'Jun'){ $mes=06; }
if ($mesmaq eq 'Jul'){ $mes=07; }
if ($mesmaq eq 'Agu'){ $mes=8; }
if ($mesmaq eq 'Sep'){ $mes=9; }
if ($mesmaq eq 'Oct'){ $mes=10;}
if ($mesmaq eq 'Nov'){ $mes=11;}
if ($mesmaq eq "Dec"){ $mes=12;}

$dbh = DBI->connect($db, $username, $pass);
$sql = qq{ SELECT * FROM servicios_contratados WHERE estado = 'A' AND
descripcion_servici! != 'CORREO' };
$dbh->do($sql);
$sth = $dbh->prepare($sql);
$sth->execute;

$fecha1= $values[4].'-'.$mes.'-'.'01';
$fecha2= $values[4].'-'.$mes.'-'.$values[2];
#print $fecha1;

while(@array = $sth->fetchrow_array){
$contrato = $array[1];
$ip = $array[8];
#print "SELECT * FROM facturas WHERE id_contrato = '$contrato' AND
estado = 'I' AND DATE_FORMAT(fecha_factura + INTERVAL 5
#DAY, '%y/%m/%d') <=DATE_FORMAT(NOW(), '%y/%m/%d')";
#print "\n";
$sq2 = qq{ SELECT * FROM facturas WHERE id_contrato = '$contrato' AND
estado = 'I' AND DATE_FORMAT(fecha_factura + INTERVAL 5 DAY, '%y/%m/%d')
<= DATE_FORMAT(NOW(), '%y/%m/%d')};
$dbh->do($sq2);
$sth1 = $dbh->prepare($sq2);
$sth1->execute;

while (@array2=$sth1->fetchrow_array){
$numero_factura = $array2[1];
$idfactura = $array2[0];
#print $numero_factura;

#print "INSERT INTO inactivacion_servicios (ip_maquina, estado) VALUES
```

```
('$ip', 'I')";
#print "\n";
#$sq3 = qq{ INSERT INTO inactivacion_servicios (ip_maquina, estado)
VALUES ('$ip', 'I')};
#$dbh->do($sq3);
#$sth2 = $dbh->prepare($sq3);
#$sth2->execute;

`iptables -t nat -D POSTROUTING -o eth0 -s $ip/32 -j MASQUERADE`;

$sq13 = qq{ DELETE FROM nat WHERE ip='$ip' };
$dbh->do($sq13);
$sth13 = $dbh->prepare($sq13);
$sth13->execute;

#$dbh = DBI->connect($db, $username, $pass);
#$sql11 = qq{ SELECT * FROM nat WHERE ip = '$ip' };
#$dbh->do($sql11);
#$sth11 = $dbh->prepare($sql11);
#$sth11->execute;
#while(@array11 = $sth11->fetchrow_array){
#$sql12 = qq{ UPDATE servicios_contratados SET estado = 'A' WHERE estado
= 'R' AND id_servicio = '$servicios' AND
#id_contrato = '$contrato' };
#$dbh->do($sql12);
#$sth12 = $dbh->prepare($sql12);
#$sth12->execute;
#}
#print "UPDATE facturas SET estado = 'C' WHERE id_contrato = '$contrato'
AND estado = 'I' AND DATE_FORMAT(fecha_factura,
#'%y/%m/%d') <= DATE_FORMAT(NOW() + INTERVAL 5 DAY, '%y/%m/%d')";
#print "\n";
#$sq3 = qq{ UPDATE facturas SET estado = 'C' WHERE id_contrato =
'$contrato' AND estado = 'I' AND DATE_FORMAT(fecha_factura, '%y/%m/%d')
<=
#DATE_FORMAT(NOW()+ INTERVAL 5 DAY, '%y/%m/%d')};
#$dbh->do($sq3);
#$sth3 = $dbh->prepare($sq3);
#$sth3->execute;

$sq3 = qq{ UPDATE servicios_contratados SET estado = 'I' WHERE estado =
'A' AND id_contrato = '$contrato' AND descripcion_servicio != 'CORREO'};
$dbh->do($sq3);
$sth3 = $dbh->prepare($sq3);
$sth3->execute;
}
}
```

Script Corte

```
#!/usr/bin/perl

use DBI;

$db = 'DBI:mysql:facturacionisp:127.0.0.1';
$username = 'root';
$pass = 'redes';

#$now = localtime time;
#@values = split(' ',$now);
#print $values[1],"-", $values[2],"-", $values[4];
#$mesmaq = $values[1];

$dbh = DBI->connect($db, $username, $pass);
$sql = qq{ SELECT * FROM inactivacion_servicios WHERE estado = 'I' };
$dbh->do($sql);
$sth = $dbh->prepare($sql);
$sth->execute;

#$fecha1= $values[4].'-'.$mes.'-'.'01';
#print $fecha1;

while(@array = $sth->fetchrow_array){
$idac = $array[0];
$ip = $array[2];
`iptables -t nat -D POSTROUTING -o eth0 -s $ip/32 -j MASQUERADE`;
$sq13 = qq{ DELETE FROM nat WHERE ip='$ip' };
$dbh->do($sq13);
$sth13 = $dbh->prepare($sq13);
$sth13->execute;
$sq2 = qq{ UPDATE inactivacion_servicios SET estado = 'E' WHERE estado = 'I' AND
id_inactivacion = '$idac' AND ip_maquina = '$ip'};
$dbh->do($sq2);
$sth1 = $dbh->prepare($sq2);
$sth1->execute;

$sq16 = qq{ UPDATE servicios_contratados SET estado = 'I', valor_referencia_3 =
'Automatico_inactivado' WHERE estado = 'A' AND valor_referencia_2 = '$ip' AND
 descripcion_servicio != 'CORREO'};
$dbh->do($sq16);
$sth16 = $dbh->prepare($sq16);
$sth16->execute;

}
```

Script Mail

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";

use DBI;

$db = 'DBI:mysql:facturacionisp:127.0.0.1';
$username = 'root';
$pass = 'redes';

$title='Factura del Mes';
$subject='Factura del Mes';
$signo = '$';
$a = '@';
$dominio = 'bemusoft.com';

$now = localtime time;
@values = split(' ',$now);
#print $values[1],"-", $values[2],"-", $values[4];
$mesmaq = $values[1];

if ($mesmaq eq 'Jan'){ $mes=1; print "no debio entrar"; }
if ($mesmaq eq 'Feb'){ $mes=2; }
if ($mesmaq eq 'Mar'){ $mes=3; }
if ($mesmaq eq 'Apr'){ $mes=4; }
if ($mesmaq eq 'May'){ $mes=5; }
if ($mesmaq eq 'Jun'){ $mes=6; }
if ($mesmaq eq 'Jul'){ $mes=7; }
if ($mesmaq eq 'Agu'){ $mes=8; }
if ($mesmaq eq 'Sep'){ $mes=9; }
if ($mesmaq eq 'Oct'){ $mes=10;}
if ($mesmaq eq 'Nov'){ $mes=11;}
if ($mesmaq eq "Dec"){ $mes=12;}

$fecha1= $values[4].'-'.$mes.'-'.'05';
$fecha2= $values[4].'-'.$mes.'-'.$values[2];

$dbh = DBI->connect($db, $username, $pass);
$sq5 = qq{ SELECT * FROM empresas };
$dbh->do($sq5);
$sth5 = $dbh->prepare($sq5);
$sth5->execute;
while (@array5=$sth5->fetchrow_array){
$razonsocial = $array5[1];;
}

#print "SELECT * FROM facturas WHERE estado = 'I' AND
DATE_FORMAT(fecha_factura, '%y/%m/%d') = DATE_FORMAT(NOW(),
#'%y/%m/%d')";
```

```
#print "\n";

$sq2 = qq{ SELECT * FROM facturas WHERE estado = 'I' AND
DATE_FORMAT(fecha_factura, '%y/%m/%d') = DATE_FORMAT(NOW(),
'%y/%m/%d') };
$dbh->do($sq2);
$sth1 = $dbh->prepare($sq2);
$sth1->execute;
while (@array2=$sth1->fetchrow_array){
$idcliente= $array2[5];
$idcontrato = $array2[4];

#print "SELECT * FROM clientes WHERE id_cliente  = '$idcliente'";
#print "\n";

$sq3 = qq{ SELECT * FROM clientes WHERE id_cliente  = '$idcliente' };
$dbh->do($sq3);
$sth3 = $dbh->prepare($sq3);
$sth3->execute;
while (@array3=$sth3->fetchrow_array){
$idpersona= $array3[1];

#print "SELECT * FROM personas WHERE id_persona  = '$idpersona'";
#print "\n";

$sq4 = qq{ SELECT * FROM personas WHERE id_persona  = '$idpersona'};
$dbh->do($sq4);
$sth4 = $dbh->prepare($sq4);
$sth4->execute;
while (@array4=$sth4->fetchrow_array){
$nombre= $array4[4] . $array4[5];

#print "SELECT * FROM servicios_contratados WHERE id_contrato =
'$idcontrato' AND descripcion_servicio = 'CORREO'";
#print "\n";

$sq11 = qq{ SELECT * FROM servicios_contratados WHERE id_contrato =
'$idcontrato' AND descripcion_servicio = 'CORREO'};
$dbh->do($sq11);
$sth11 = $dbh->prepare($sq11);
$sth11->execute;
while (@array11=$sth11->fetchrow_array){

$cuenta=$array11[8];
print $cuenta;
print "\n";
$to=$cuenta . $a . $dominio;
print $to;
```

```
$from= 'root@mail.bemusoft.com';

$valor = $array2[7] + $array2[8] + $array2[9] - $array2[11];
#$fecha = DATE_FORMAT(fecha_registro + INTERVAL 5 DAY, '%y/%m/%d');

open(MAIL, "|/usr/sbin/sendmail -t");

## Mail Header
print MAIL "To: $to\n";
print MAIL "From: $from\n";
print MAIL "Subject: $subject\n\n";
## Mail Body

print MAIL "Estimado Cliente  ", $nombre;
print MAIL "\n\n";

print MAIL "Le recordamos que su factura ya fue emitida con un valor de:", $signo;
print MAIL $valor;
print MAIL "\n";
print MAIL "la misma que puede ser cancelada en cualquiera de nuestras oficina
hasta";
print MAIL "\n";
print MAIL "5 dias despues de la fecha de la factura";
print MAIL ", en que se realiza el corte automatico.";
print MAIL "\n";
print MAIL "Gracias";
print MAIL "\n";
print MAIL "Att";
print MAIL "\n";
print MAIL $razonsocial;

close(MAIL);

print "<html><head><title>$title</title>
</head>\n<body>\n\n";

## HTML content let use know we sent an email
print "<h1>$title</h1>\n";
print "<p>A message has been sent from $from to $to";
print "\n\n</body></html>";
}
}
}
}
```

Script Renuncia

```
#!/usr/bin/perl
```

```
use DBI;

@datos = ("RULE=");

$db = 'DBI:mysql:facturacionisp:127.0.0.1';
$username = 'root';
$pass = 'redes';

#print "SELECT * FROM contratos WHERE id_contrato = '$contrato'";
$dbh = DBI->connect($db, $username, $pass);
$sq3 = qq{ SELECT * FROM contratos WHERE estado = 'X' };
$dbh->do($sq3);
$sth3 = $dbh->prepare($sq3);
$sth3->execute;
while(@array3 = $sth3->fetchrow_array){
$idcontrato = $array3[0];
$cuenta = $array[1];

$dbh = DBI->connect($db, $username, $pass);
$sq6 = qq{ SELECT * FROM servicios_contratados WHERE id_contrato =
'$idcontrato' AND descripcion_servicio != 'CORREO' };
$dbh->do($sq6);
$sth6 = $dbh->prepare($sq6);
$sth6->execute;
while(@array6 = $sth6->fetchrow_array){
$ip = $array6[8];
`iptables -t nat -D POSTROUTING -o eth0 -s $ip/32 -j MASQUERADE`;
$linea=$datos[0].$ip."/32";
}

$dbh = DBI->connect($db, $username, $pass);
$sq6 = qq{ SELECT * FROM servicios_contratados WHERE id_contrato =
'$idcontrato' AND valor_referencia_3 = 'CORREO' AND valor_referencia_2 !=
'NULL' };
$dbh->do($sq6);
$sth6 = $dbh->prepare($sq6);
$sth6->execute;
while(@array6 = $sth6->fetchrow_array){
$correo = $array6[8];
system("userdel -rf $correo");
}

$dbh = DBI->connect($db, $username, $pass);
$sq15 = qq{ SELECT * FROM productos WHERE descripcion = '$cuenta' };
$dbh->do($sq15);
$sth15 = $dbh->prepare($sq15);
$sth15->execute;
```

```
while(@array15 = $sth15->fetchrow_array){
$nombrefile = $array15[18];
$ruta= "/etc/sysconfig/cbq/".$nombrefile;
`more $ruta | grep -v $linea > /etc/sysconfig/cbq/test2`;
`rm -rf $ruta`;
`mv /etc/sysconfig/cbq/test2 $ruta`;
}

$dbh = DBI->connect($db, $username, $pass);
$sq6 = qq{ UPDATE contratos SET estado= 'I' WHERE estado = 'X' AND
id_contrato = '$idcontrato' };
$dbh->do($sq6);
$sth6 = $dbh->prepare($sq6);
$sth6->execute;
}
```